REDEFINING "COMPUTATION" ITSELF

THE HYPERDIMENSIONAL ALGORITHM

A NEW DEFINITION OF "COMPUTATION"

超维算法

核心要点：

1 多维结构
数据不是单一输入输出，
而是多起点、多结果的结构网络

2 非线性因果
结果可以先于原因，
多路径并行存在

3 随机有序
随机性与结构性同时存在于系统内部

4 一数多果
同一数据可对应多个结果，
不依赖单一计算路径

5 入口触发
通过改变入口而非修改数据，
实现不同结果输出

MULTI-DIMENSIONAL STRUCTURE
Data as nodes. Not just inputs or outputs, but multiple origins and multiple results.

NON-LINEAR CAUSALITY
Results can precede causes. Multiple paths. Recursive and interconnected.

RANDOM YET ORDERED
Randomness and order coxrist as an intrinsic part of the algorithmic structure.

ONE DATA, MANY RESULTS
The same metadata can correspond to different results through different structural entrances.

ENTRANCE-BASED ACTIVATION
Change the entrance, not the data. Activate different states and outcomes.

A FOUNDATIONAL FRAMEWORK
FOR HYPERDIMENSIONAL COMPUTATION
AND A NEW SCIENTIFIC PARADIGM

Contents

[极限算法]超维算法 1
引言：为什么要重新讨论“算法” 1
第一部分：当下对“算法”的标准定义 2
第二部分：“超维算法”的定义 3
第三部分：超维算法与传统算法的根本差异 5
第四部分：果因论哲学 6
第五部分：一个日常例子——裤子与腰部折叠 7
第六部分：不修改元数据的数据观 8
第七部分：重新定义“超算” 9
第八部分：实证基础——多领域系统验证与新范式判定 10
第九部分：常见误解澄清 11
第十部分：超维算法与人类计算史的关系 12
第十一部分：总结——这不是优化，而是重定义 12
附录：本文的边界与开放问题 14
参考文献性说明 15
[Limit Algorithm] The Hyperdimensional Algorithm 16
Introduction: Why Re-discuss "Algorithm" 17
Part One: The Standard Definition of "Algorithm" Today 18
Part Two: Definition of the "Hyperdimensional Algorithm" 20
Part Three: Fundamental Differences Between HDA and Traditional Algorithms 23
Part Four: The Effect-to-Cause Philosophy 25
Part Five: An Everyday Example—Pants and Waistband Folding 26
Part Six: A Data View of Unmodified Metadata 29
Part Seven: Redefining "Super-computation" 31

Part Eight: Empirical Basis—Multi-domain Systematic Validation and New Paradigm Determination 32
Part Nine: Clarification of Common Misunderstandings 33
Part Ten: The HDA in the History of Human Computation 35
Part Eleven: Conclusion—This is Not Optimization, but Redefinition 36
Appendix: Boundaries and Open Questions of this Paper 39
Bibliographic Notes 41
[Algorithme Limite] L'algorithme hyperdimensionnel 42
Introduction : Pourquoi rediscuter de l'« algorithme » 43
Première partie : La définition standard de l'« algorithme » aujourd'hui 44
Deuxième partie : Définition de l'« algorithme hyperdimensionnel » 46
Troisième partie : Différences fondamentales entre l'AHD et les algorithmes traditionnels 50
Quatrième partie : La philosophie de l'Effet-vers-la-Cause 52
Cinquième partie : Un exemple quotidien – Le pantalon et le pliage de la ceinture 53
Sixième partie : Une vue des données fondée sur des métadonnées inchangées 57
Septième partie : Redéfinir le « supercalcul » 59
Huitième partie : Fondement empirique – Validation systématique multi-domaines et détermination d'un nouveau paradigme 60
Neuvième partie : Clarification des malentendus courants 62
Dixième partie : L'AHD dans l'histoire du calcul humain 63
Onzième partie : Conclusion – Ce n'est pas une optimisation, mais une redéfinition 65
Annexe : Frontières et questions ouvertes de cet article 68
Notes bibliographiques 69
[Algoritmo Límite] El Algoritmo Hiperdimensional 71
Introducción: Por qué rediscutir el «algoritmo» 72
Primera parte: La definición estándar de «algoritmo» hoy 73
Segunda parte: Definición del «Algoritmo Hiperdimensional» 75

Tercera parte: Diferencias fundamentales entre el AHD y los algoritmos tradicionales 79
Cuarta parte: La filosofía del Efecto-hacia-la-Causa 81
Quinta parte: Un ejemplo cotidiano – El pantalón y el pliegue de la cintura 82
Sexta parte: Una visión de los datos basada en metadatos inalterados 86
Séptima parte: Redefiniendo el «supercálculo» 87
Octava parte: Fundamento empírico – Validación sistemática en múltiples dominios y determinación de un nuevo paradigma 88
Novena parte: Aclaración de malentendidos comunes 90
Décima parte: El AHD en la historia del cálculo humano 92
Undécima parte: Conclusión – Esto no es optimización, sino redefinición 93
Apéndice: Fronteras y preguntas abiertas de este artículo 96
Notas bibliográficas 97
[極限アルゴリズム] 超次元アルゴリズム —「計算」そのものの再定義 99
序論：なぜ「アルゴリズム」を再検討するのか 100
第一部分：現在の「アルゴリズム」の標準的定義 101
第二部分：「超次元アルゴリズム」の定義 103
第三部分：超次元アルゴリズムと従来アルゴリズムの根本的差異 106
第四部分：「結果から原因へ」の哲学 108
第五部分：日常的な例——ズボンとウエストの折り曲げ 109
第六部分：メタデータを変更しないデータ観 112
第七部分：「超計算」の再定義 114
第八部分：実証的基盤——複数領域でのシステム検証と新パラダイム判定 115
第九部分：よくある誤解の解消 116
第十部分：人類の計算史における超次元アルゴリズムの位置づけ 118
第十一部分：結論——これは最適化ではなく、再定義である 119
付録：本稿の境界と未解決問題 121
参考文献的説明 123
نفسها "الحوسبة" تعريف إعادة 124

مقدمة: "الخوارزمية" مناقشة نعيد لماذا 125
الجزء الأول: التعريف الحالي المعياري للخوارزمية 126
الجزء الثاني: تعريف الخوارزمية فائقة الأبعاد 127
الجزء الثالث: الاختلافات الجوهرية بين الخوارزميات التقليدية والخوارزمية فائقة الأبعاد 129
الجزء الرابع: فلسفة الأثر قبل السبب 131
الجزء الخامس: مثال يومي – البنطلون وطي حزام الخصر 131
الجزء السادس: نظرة بيانات تقوم على بيانات غير وصفية متغيرة 134
الجزء السابع: إعادة تعريف "الفائقة الحوسبة" 135
الجزء الثامن: الأساس التجريبي – التحقق متعدد المجالات وتحديد النموذج الجديد 136
الجزء التاسع: توضيح المفاهيم الخاطئة الشائعة 137
الجزء العاشر: تاريخ الحوسبة البشرية في الأبعاد فائقة الخوارزمية 138
الجزء الحادي عشر: استنتاج – هذا ليس تحسيناً، بل إعادة تعريف 139
ملحق: حدود هذه الورقة والأسئلة المفتوحة 141
إيضاحات ببليوغرافية 141
[Algoritmo Limite] O Algoritmo Hiperdimensional 143
Introdução: Por que rediscutir "algoritmo" 144
Primeira Parte: A Definição Padrão de "Algoritmo" Hoje 145
Segunda Parte: Definição do "Algoritmo Hiperdimensional" 147
Terceira Parte: Diferenças Fundamentais entre o AHD e os Algoritmos Tradicionais 151
Quarta Parte: A Filosofia do Efeito-para-Causa 153
Quinta Parte: Um Exemplo Quotidiano – As Calças e a Dobra da Cintura 154
Sexta Parte: Uma Visão dos Dados Baseada em Metadados Inalterados 157
Sétima Parte: Redefinindo a "Supercomputação" 159
Oitava Parte: Fundamento Empírico – Validação Sistemática em Múltiplos Domínios e Determinação de um Novo Paradigma 160
Nona Parte: Esclarecimento de Mal-entendidos Comuns 162
Décima Parte: O AHD na História do Cálculo Humano 164
Décima Primeira Parte: Conclusão – Isto não é Otimização, mas Redefinição 165

Apêndice: Fronteiras e Questões em Aberto deste Artigo 167
Notas Bibliográficas 168
[Grenzalgorithmus] Der hyperdimensionale Algorithmus 170
Einleitung: Warum wir „Algorithmus“ neu diskutieren sollten 171
Teil Eins: Die heutige Standarddefinition von „Algorithmus“ 172
Teil Zwei: Definition des „hyperdimensionalen Algorithmus“ 174
Teil Drei: Grundlegende Unterschiede zwischen HDA und traditionellen Algorithmen 178
Teil Vier: Die Wirkung-zu-Ursache-Philosophie 180
Teil Fünf: Ein alltägliches Beispiel – Die Hose und das Falten des Bundes 182
Teil Sechs: Eine Datenperspektive unveränderter Metadaten 185
Teil Sieben: Neudefinition von „Superberechnung“ 187
Teil Acht: Empirische Grundlage – Systematische Validierung in mehreren Bereichen und Bestimmung eines neuen Paradigmas 188
Teil Neun: Klarstellung häufiger Missverständnisse 190
Teil Zehn: Der HDA in der Geschichte der menschlichen Berechnung 192
Teil Elf: Schlussfolgerung – Dies ist keine Optimierung, sondern eine Neudefinition 193
Anhang: Grenzen und offene Fragen dieses Papiers 196
Bibliographische Hinweise 197
[Предельный алгоритм] Гипермерный алгоритм 199
Введение: Зачем заново обсуждать «алгоритм» 200
Часть первая: Стандартное определение «алгоритма» сегодня 201
Часть вторая: Определение «гипермерного алгоритма» 202
Часть третья: Фундаментальные различия между ГМА и традиционными алгоритмами 206
Часть четвертая: Философия Следствия-к-Причине 207
Часть пятая: Повседневный пример – Брюки и складывание пояса 209
Часть шестая: Взгляд на данные, основанный на неизменности метаданных 212

Часть седьмая: Переопределение «супервычисления» 213
Часть восьмая: Эмпирическая основа – Систематическая валидация в нескольких областях и определение новой парадигмы 214
Часть девятая: Разъяснение распространенных заблуждений 216
Часть десятая: ГМА в истории человеческих вычислений 217
Часть одиннадцатая: Заключение – Это не оптимизация, а переопределение . 218
Приложение: Границы и открытые вопросы данной статьи 220
Библиографические примечания 221
[극한 알고리즘] 초차원 알고리즘 223
서론: 왜 '알고리즘'을 다시 논의하는가 224
제 1 부: 현재 '알고리즘'의 표준적 정의 225
제 2 부: '초차원 알고리즘'의 정의 226
제 3 부: 초차원 알고리즘과 전통적 알고리즘의 근본적 차이 229
제 4 부: '결과-원인' 철학 231
제 5 부: 일상적 사례 – 바지와 허리 접기 232
제 6 부: 메타데이터를 변경하지 않는 데이터관 235
제 7 부: '초계산'의 재정의 237
제 8 부: 실증적 기반 – 다영역 시스템 검증과 새 패러다임 판정 238
제 9 부: 흔한 오해 해소 239
제 10 부: 인류 계산사 속의 초차원 알고리즘 241
제 11 부: 결론 – 이것은 최적화가 아니라 재정의이다 242
부록: 본 논문의 경계와 미해결 문제 244
참고문헌적 설명 246

[极限算法]超维算法

一种对“计算”本身的重新定义

作者：巫朝晖 JEFFI CHAO HUI WU

摘要

本文提出“超维算法”这一新概念，并将其作为对传统算法定义边界的一次结构性重审。传统算法通常建立在输入、步骤、规则、输出的线性框架之上，强调有穷性、确定性、能行性、可行性以及清晰的输入输出边界。现代超级计算机虽然拥有极高算力，但其底层逻辑仍然主要是在更大规模上执行既定算法范式。本文认为，真正值得讨论的“超算”并不只是算力规模的增长，而是计算范式本身的改变。超维算法不是传统算法的加速版，也不是数学算法的扩展，而是一种多维、叠加、随机而有序的结构性计算观。在这一结构中，数据不再只是被动输入或终点输出，而是同时具备多个起点与多个结果的节点属性；结果不再只是终点，也可以成为新的入口；元数据不必被反复修改，而可以通过结构入口、关系变化与条件激活，对应不同状态与结果。本文以“果因论”哲学和“裤子腰部折叠”这一日常例子，说明超维算法如何通过改变结构入口，使目标结果直接成立，从而减少重复计算，甚至绕开原有计算路径。本文旨在为“超维算法”建立初步定义、结构边界与思想基础，为未来进一步展开超维计算、极限算法与新科学体系提供原始理论节点。按“存在且成立即为新范式”的标准，超维算法作为一种已被多领域系统验证的新计算结构范式，在当下已经成立。

关键词： 超维算法；极限算法；超维计算；果因论；元数据；计算范式；结构入口；非线性因果；新科学

引言：为什么要重新讨论“算法”

我提出“超维算法”，并不是为了给现有算法换一个更大的名称，也不是为了把传统算法包装成某种新奇概念。恰恰相反，我要讨论的是一个更根本的问题：人类当下对“算法”的理解，是否已经被限制在一个过于狭窄的框架之中？

今天人们一说算法，通常想到的是数学公式、程序代码、逻辑步骤、输入输出、模型训练、数据处理、路径优化、搜索排序、自动推荐和人工智能推理。这些当然都是算法的重要形式，也支撑了现代计算机、互联网、数据库、人工智能、超级计算、工业系统和信息社会的运行。但是，如果把算法只理解为“一套有序步骤”，只理解为“从输

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

入开始，经过规则处理，最后得到输出”的过程，那么这种理解本身就已经把计算限制在一个低维框架之内。

我们生活在一个被算法支配的时代。从搜索引擎到推荐系统，从天气预报到人工智能，算法的边界就是人类智能的边界。但有一个问题很少被追问：算法只能是这样吗？为什么计算必须遵循“输入→步骤→输出”的线性模式？为什么相同的问题每次都要从头计算？本文试图挑战这些默认假设，提出一种完全不同的计算范式——超维算法。按“存在且成立即为新范式”的标准，超维算法作为一种已被多领域系统验证的新计算结构范式，在当下已经成立。

需要特别说明的是，本文提出的“超维算法”与学术界已有近三十年发展历史的“超维计算”（Hyperdimensional Computing, HDC）是完全不同的概念。HDC 使用极高维度的随机二进制向量进行编码和运算，追求更高效的表示和计算，但仍属于“输入→处理→输出”的线性范式之内。超维算法则完全不同：它追问的是结果是否可以通过结构直接成立，是否可以绕过计算路径本身，是否可以不修改元数据而适配多个结果。两者名称相近，但内涵相反。超维算法不是对算法的升级，而是对“算法是否必须存在”这一前提的重新提问。

第一部分：当下对“算法”的标准定义

在主流计算机科学、数学和工程体系中，算法有一个被长期接受的基本定义：算法是一个定义良好的、由有限步骤构成的计算过程，它接受一个或多个输入，经过确定性的规则转换，在有限时间内产生一个或多个输出。这一理解不是某一个人的随意观点，而是现代计算文明长期形成的基础共识。它支撑了软件、硬件、数据库、网络系统、人工智能模型和超级计算中心的底层逻辑。无论算法表现得多么复杂，其核心仍然围绕输入、规则、步骤和输出展开。

这一标准算法观念可以拆解为几个核心特征。

第一，有穷性。算法必须在有限步骤内终止，不能无限循环，也不能永远运行下去。一个永远不会停下来的过程，即使形式上可以被描述，也很难被视为有效算法。所有科学计算程序、数据处理流程，都有明确的结束条件。

第二，确定性。相同的输入，在相同条件下，应当产生相同的输出。即使某些算法引入随机数，也通常是受控的随机、可复现的随机，或者在统计意义上可解释的概率机制，而不是完全不可把握的内在随机。数值模拟的结果必须可复现，这是科学计算的基本要求。

第三，明确的输入输出边界。输入在前，处理在中，输出在后，起点和终点具有清楚的结构边界。你给算法什么数据，它处理完之后给你什么结果，这个边界是分明的。输入在前，处理在中，输出在后，顺序不可颠倒。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

第四，能行性。算法中的每一步都必须是实际可执行的操作，不能包含无法执行、无法描述、无法验证的跳跃。每一步都必须是人类能用纸笔或者计算机能实际执行的基本操作——加减乘除、逻辑判断、数据读写等。

第五，可行性。一个算法即使在理论上成立，如果消耗的时间、算力、内存或存储资源完全不可接受，在工程意义上也难以成为有效算法。一个算法如果理论上正确但需要几亿年才能算完，在工程上不被视为可行的算法。

这一整套算法观念，最终可以追溯到图灵机模型。图灵机作为现代计算理论的核心抽象，给出了“可计算”的基本边界。无论今天的计算机多么强大，无论芯片多么先进，无论超级计算中心规模多么庞大，其底层逻辑仍然没有真正脱离这一条路径：输入、规则、步骤、输出。现代超级计算机只是把这一过程用更大的硬件规模、更高的并行度、更复杂的系统调度加速执行。也就是说，传统语境中的“超算”，主要仍然是算力规模的扩张，而不一定是计算范式的根本改变。算力可以成千上万倍增长，但如果底层逻辑仍然是“输入、步骤、输出”，那么它仍然处在传统算法范式之内。

第二部分：“超维算法”的定义

我所提出的“超维算法”，正是在这一背景下出现的。它不是传统算法的改良版，不是更快的算法，不是更复杂的数学公式，也不是某种更高级的程序技巧。它是一种完全不同的结构性理解。

首先需要解释“超维”一词的含义。“超维”在这里有两层意思。第一层，它不局限于一维的线性步骤序列，而是允许多个维度同时展开。第二层，它试图超越传统算法对“计算”的定义边界——算法不再必须是数学的、有序的、确定性的。传统算法像是在一条单行道上开车，你必须从 A 点出发，经过 B、C、D 才能到达 Z。超维算法不认为计算必须是一条单行道。它认为计算可以是一个网络、一个场、一个多维结构，其中任意节点都可能是入口，任意节点都可能是出口。

在我的新科学体系中，算法不一定是数学算法，不一定是单一有序计算，不一定必须沿着固定步骤运行，也不一定必须严格服从“输入、处理、输出”的线性模式。超维算法是一种多维、叠加、随机而有序的结构。在这种结构中，每一个数据都不是孤立的输入，也不是固定的输出，而是同时具备多重角色：它既可以成为多个结果的起点，也可以成为多个起点共同作用之后形成的结果。

换句话说，传统算法把数据放在流程里，而超维算法把数据放在结构里。传统算法中的数据，通常被分成输入数据、中间数据和输出数据。每一种数据都有清楚的位置和角色。输入就是输入，结果就是结果，中间过程就是中间过程。可是，在超维算法中，一个数据并不是只有一个身份。它可能在一个方向上是结果，在另一个方向上又

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

是起点；它可能在一个结构中是被调用的对象，在另一个结构中又成为触发新结果的入口。数据不再只是被动材料，而是一个多维关系节点。

这正对应超维算法的核心：每个数据都是多个结果的起点、也是多个起点的结果。传统算法以“有序步骤”产生单一结果；而超维算法更接近一种多维状态结构，在随机与有序的叠加中，结果并非被计算生成，而是在结构中自然显现。在这一体系中，数据既是起点，也是结果的组成部分。

为了更清晰地呈现超维算法的核心特征，我将其拆解为以下五个方面。

第一，非数学性。主流算法本质上是数学的——可以用符号逻辑和数学公式完整描述。超维算法不一定是数学的。它可能基于物理原则、几何关系、信息论新范式，甚至意识原则。计算不一定通过“数学运算”完成，可能通过高维空间中的几何关系自然呈现。举例来说，肥皂泡的形状是由表面张力最小化原理决定的。这不是一个“数学算法”在计算——是物理本身在“计算”。超维算法试图理解这种“计算”，而不是用数学公式去模拟它。

第二，多维与叠加。传统算法在单一维度上执行有序步骤，每一步只有一个状态。超维算法允许多个维度同时存在、多种状态叠加共存。算法不沿着一条路径走下去，而是在一个多维的状态场中展开。结果不是“算”出来的，而是从这个场中“显现”出来的。举例来说，一片雪花在空中形成时，它并没有一个“算法”在告诉每个水分子该去哪里。水分子的排列是温度、湿度、气流等多个维度共同作用的结果。雪花的形状“显现”出来，而不是被“计算”出来。

第三，随机而有序。传统算法中的随机是工具的、外部的、可控的扰动，算法本身是确定的。超维算法中的随机是内禀的、结构性的。随机与有序同时存在、协同工作，共同构成算法的本质。这不是“有随机性的算法”，而是“随机本身就是算法的有机组成部分”。举例来说，森林的生态结构——树木的分布看起来是随机的，但整体上又呈现出有序的密度分布和物种关系。那个“随机”不是 bug，而是生态结构正常运作的前提。

第四，数据的多重角色。传统算法中，数据的角色是单一的——输入就是输入、输出就是输出、中间结果就是中间结果。超维算法中，每个数据同时是多个结果的起点，也是多个起点的结果。一个数据节点可以向下游展开出多个结果路径，也可以从上游被多个原因汇聚而来。数据不再是线性流程中的一个点，而是一个网络中的交汇节点。举例来说，一个人的身高。在传统算法中，身高是一个“输入”——你把它输进去，得到体重预测、服装尺码等“输出”。但身高同时也是一个“结果”——它是由基因、营养、运动等多个“起点”共同决定的。在超维算法中，身高这个数据同时是起点和结果，而不是二选一。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

第五，元数据不变，关系可变。传统算法在面对不同问题时，要么修改数据本身，要么重新计算一遍。超维算法不修改元数据——数据的本质属性保持不变。改变的是入口、是解释方式、是数据之间的关系。同一个元数据结构，通过不同的入口激活，可以呈现出完全不同的结果。这意味着：算一次，用无数次。举例来说，同样一段文字。你可以把它当作诗歌来读，也可以把它当作密码来解码，也可以把它当作历史文献来分析。文字本身没有变——你只是改变了“入口”。超维算法追求的就是这种“同一数据，多入口，多结果”的能力。

从数据结构角度看，不对元数据本身进行修改，而是通过不同的入口与条件，使其在同一结构中对应多个起点与结果。数据不再被反复加工，而是在保持结构不变的前提下，通过不同入口被激活，从而呈现出不同结果。传统是“为不同结果加工数据”，而超维算法是“用同一数据承载多种结果可能”。

第三部分：超维算法与传统算法的根本差异

基于以上定义，超维算法与传统算法在多个核心维度上存在根本差异。以下逐一展开。

在本质属性上，传统算法是数学的、形式的，可以完全用符号逻辑和数学公式描述，本质上是一个数学客体。超维算法不一定是数学的，它可能基于物理、几何、信息甚至意识原则，它不是数学的一个子集，而是一个更广阔的范畴。这一差异可以概括为：从“数学客体”到“宇宙法则”。

在时间秩序上，传统算法遵循单一、有序、线性的时间秩序，步骤 A 到 B 到 C，严格串行，或者可拆分为并行的有序子步骤，时间箭头不可逆转。超维算法没有固定的步骤序列，它是多维、叠加、随机而有序的，结果可能与执行的“顺序”无关，因为根本不存在传统意义上的“顺序”。这一差异可以概括为：从“线性时间”到“高维同时性”。

在因果关系上，传统算法遵循确定性因果链，给定相同的输入和初始状态，输出永远唯一，原因在前，结果在后，这是一个不可逆的单向箭头。超维算法遵循叠加态因果，每个数据是多个结果的起点，也是多个起点的结果。这就是我的“果因论”哲学——可以先有结果，然后才有原因。结果不是终点，而是可以成为新的起点。这一差异可以概括为：从“单因单果”到“全互联因果网”。

在数据结构上，传统算法遵循输入到处理再到输出的单向流动结构，数据角色单一，有清晰的边界，输入数据从头到尾是输入，结果数据永远是结果。超维算法中数据角色多重，同一个数据同时是结果和起点，没有绝对的起点和终点，只有网络中的节点。这一差异可以概括为：从“单向流动性”到“递归共生性”。

在计算模式上，传统算法每次遇到问题都从头计算，即使已经算过一千次类似的问题、即使答案已经知道，第一千零一次仍然要走完整个流程，这是“无状态”的计算。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

超维算法如果有相应的结果就不必再跑一次，通过一个结果可以推理出多个起点，可以反向展开原因路径，计算是有状态的、有记忆的、可复用的。这一差异可以概括为：从“每次重算”到“一次复用”。

在随机性的角色上，传统算法中的随机是伪随机或外部注入的，随机数只是工具，用于采样、近似或优化，算法本身是确定的。超维算法中的随机是内禀的、建构性的，随机是有机组成部分，与有序性共存、协同，不是“算法里有随机”，而是“随机是算法之所以能运作的原因之一”。这一差异可以概括为：从“工具性随机”到“建构性随机”。

在对资源的理解上，传统算法追求在给定资源下算得更快、更准，资源是约束条件，算法的目标是“在约束内完成计算”。超维算法追求通过结构设计，让计算本身变得不再必要，不是“更快地算”，而是“绕过算”，资源不是用来“消耗”的，而是用来“设计入口”的。这一差异可以概括为：从“资源约束下的优化”到“结构设计中的消除”。

第四部分：果因论哲学

基于以上对比，我需要进一步展开“果因论”哲学。这是超维算法的核心思想基础之一。

传统思维通常认为，原因在前，结果在后；先有因，后有果；先输入，再输出。这个观念在算法中的体现是：你必须从起点开始，一步一步走到终点。即使你知道答案是什么，你也不能直接“用”那个答案——因为你没有“证明路径”。

在我的结构中，结果并不一定只是终点。一个结果一旦出现，它就可以成为新的起点，继续参与新的结构生成。结果可以反向参与原因的形成。可以通过结果推理出多个可能的起点。因与果不再是单向排列，而是形成循环关系、网络关系、多维关系。

换句话说，传统算法问的是“给定原因，如何得到结果”。超维算法问的是“给定结果，如何反推或构造原因”。

需要特别澄清的是，“先有结果，然后才有原因”并不是在否定因果关系，也不是在主张“结果凭空产生”。而是在说明：在更高维的结构中，因与果可以互为入口，互相转换。就像在接下来的裤子例子中，“能穿且不拖地”这个结果先被确定，然后我反向找到了“折腰部”这个操作点，这个操作点对应着结构中的“原因”层面。不是结果没有原因，而是结果成为了发现原因的新入口。

传统算法世界里的一个根本假设是：时间是单向的，你必须从初始状态一步一步算到终态。即使你知道答案是什么，你也不能直接“用”那个答案，因为你没有证明路径。而超维算法挑战的正是这个假设：如果结果已知，原因可以被反推出来；同一个结果可以对应多个原因路径；计算不再是从起点走到终点，而是从终点展开所有可能的起点。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

第五部分：一个日常例子——裤子与腰部折叠

为了更直观地理解上述抽象差异，我用一个日常例子来说明。

一个人买了一条腰部是松紧带的新裤子，裤腿长了一点，拖地不方便。

传统做法是：发现裤腿长了，卷起一节裤腿，用针线缝上，然后试穿。如果孩子长高了，裤子又短了，缝死的裤腿放不下来，这条裤子就废了。下次再买新裤子，再缝一遍。这种方法看起来很自然，因为问题表面上出现在裤腿，所以就去处理裤腿。它对应的是传统算法的思维：发现问题，定位表象，沿着表象所在的位置进行处理，最后得到一个结果。裤腿长了，就改裤腿；数据出错了，就改数据；路径不合适，就沿路径继续修补。

我的做法不同。我不是去改裤脚，而是在腰部折一下，马上就可以穿。这个动作非常简单，但它背后的结构逻辑完全不同。我没有裁剪裤腿，没有缝死裤脚，没有改变裤子的原始长度，也没有破坏裤子的元数据。裤子的腰围、裤长、版型、布料都没有发生本质改变。我只是改变了腰部这个结构入口，使裤腿在实际穿着中自然变短。腰部在这里不是一个普通局部，而是整个穿着状态的全局控制点。通过调整这个控制点，下游问题直接消失。

更重要的是，这种方法是可逆的、可调的、可复用的。对于正在长身体的孩子，这种方式尤其明显。如果孩子现在身高不够，裤腿稍长，就在腰部折一下；过一段时间孩子长高了，就把腰部放下来一点；再长高，就完全放开。整条裤子的原始结构没有被破坏，却可以适应孩子不同阶段的身高。传统方法如果把裤脚缝死，孩子一长高，裤子可能就短了；如果剪掉，更无法恢复。我的方法则保持元数据不变，让同一结构适配多个状态。

这个例子揭示了几个重要的结构差异。

传统做法对应传统算法的逻辑：问题在裤腿，所以必须修改裤腿，沿着“原因到结果”的路径一步步操作，每一步都不能跳过。一次解决一个问题，不可逆，下次遇到同样问题还要再来一遍。我的做法对应超维算法的逻辑：目标是不拖地，我不去解决“裤腿长”这个表面原因，而是找到了一个全局控制点——腰部。折一下腰部，裤腿自动变短，问题瞬间消失。我不修改任何元数据，腰部折痕只是一个临时的、可逆的、动态的折叠状态。

从数据角度看，传统做法修改了元数据——缝短裤腿，原始数据丢失。我的做法不修改任何元数据，原始尺寸完好，只是增加了一个临时的、可逆的、动态的折叠状态。腰部折痕没有创造新数据，也没有破坏旧数据，它只是重新排列了数据之间的关系——缩短了腰部有效周长，间接改变了裤腿的有效长度。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

在这个例子中，“裤腿不拖地”是目标结果。传统方法从“裤腿长”这个原因出发，去修改裤脚，最后得到“不拖地”的结果。而我的方法是先明确“不拖地”这个结果，然后反向寻找结构入口，最后发现腰部折一下就能让结果成立。这就是“果因论”的实际体现。不是必须从原因一步步走向结果，而是可以从结果反向决定结构，使原本的路径不再必要。

这个例子还回答了关于“元数据”的问题。什么是元数据？在裤子例子里，元数据是裤子的原始尺寸：腰围、裤长、版型、布料。这是裤子的“本质属性”。临时数据是腰部那个折痕，它没有改变裤子的任何原始尺寸，只是临时折叠了一段。传统做法修改了元数据——裤长被永久缩短了，原始数据丢失。我的做法不修改任何元数据——裤子的原始尺寸完好无损，只是增加了一个临时的、可逆的、动态的折叠状态。

腰部的原始尺寸作为元数据，同时是“多个结果的起点”：它同时支撑着正常穿、腰部折一下穿、腰部折两下穿等多种穿着状态。它也是“多个起点的结果”：它是由布料选择、剪裁方式、设计意图等多个起点共同决定的。腰部折痕作为临时数据，没有创造新数据，也没有破坏旧数据，只是重新排列了数据之间的关系。

这并不是普通的小技巧，而是一种结构思维。很多人看到裤腿长，就被裤腿这个表象吸引住了，于是所有处理都围绕裤腿展开。但如果从整体结构看，裤腿长只是一个下游表现，腰部位置变化可以影响整条裤子的实际落点。也就是说，问题并不一定要在问题出现的位置解决。很多低维问题，真正的控制点往往在更高一层结构中。传统算法容易沿着问题表面继续计算，而超维算法寻找的是结构入口。

这也解释了为什么超维算法不是更复杂，而可能更简单。真正高层级的结构，往往不是把问题变复杂，而是让复杂问题在正确入口处变简单。传统算法面对复杂问题，可能增加步骤、增加模型、增加算力、增加存储、增加训练时间。超维算法则要寻找一个结构节点，一旦这个节点被正确激活，原本复杂的路径就可能失效。所谓“绕过路径”，不是偷懒，而是重新定义路径是否必要。

这个例子中：传统是“沿路径修正结果”，而超维算法是“改变入口，让路径整体失效”。常规方法是在结果端不断修补，而另一种方式是直接改变结构入口，使原本需要多次处理的问题，在起点处一次性被重构。传统算法的主流形态是“从起点出发，沿路径得到结果”，而超维算法中“结果本身可以成为路径入口，并参与新的结构生成”。传统是“一次结构对应一次结果”，而超维算法是“一个结构承载多个结果状态”。

第六部分：不修改元数据的数据观

从数据角度而言，超维算法有一个非常重要的原则：不修改元数据，使它适用于多个起点或结果。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

元数据是数据的原始属性、基础结构和本体信息。传统处理方式面对不同问题时，往往会修改数据、复制数据、加工数据、重新计算数据，或者为不同结果建立不同路径。这样做当然可以解决问题，但代价是数据被不断加工，路径被不断重复，系统复杂度不断增加。

而超维算法强调，在尽量不修改元数据的前提下，通过改变入口、关系、条件和激活方式，使同一个元数据结构对应多个起点和多个结果。数据本体不动，关系发生变化；基础结构不变，呈现方式改变；元数据保持完整，却可以适应不同状态。

这就是我所说的“算一次，用无数次”。这里的“算一次”不是简单的缓存结果，而是指一个结构一旦成立，它就不再需要为每一个状态重新建立完整路径。同一个数据结构可以通过不同入口被激活，呈现出不同结果。它不是“为不同结果加工数据”，而是“用同一数据承载多种结果可能”。这也是超维算法区别于传统算法的核心之一。传统算法处理的是路径上的数据，超维算法处理的是数据、入口、关系、结果之间的整体结构。

在最小结构上，超维算法可以被理解为：在不修改元数据的前提下，通过结构入口变化，使同一数据节点对应多个结果路径的系统。这个定义并不追求把超维算法立刻收缩为一个传统数学公式，而是先建立它的结构边界。它不是一个单线性函数，不是一个固定公式，也不是一种简单缓存机制。它是一种关系结构：元数据保持稳定，入口可以变化，条件可以变化，关系可以变化，结果可以多向呈现。正是在这个结构中，计算不再只是执行步骤，而是激活关系。

从数据角度而言，“不修改元数据，使它适用于多个起点或结果”的本质是：数据本体保持不变，变化发生在“解释方式/使用入口”。传统结构是问题变化导致数据或路径变化；超维算法是问题变化导致解释结构变化，而非数据变化。

第七部分：重新定义“超算”

在这个体系中，我需要澄清一个词——“超算”。

我所说的“超算”，并不是通常意义上的超级计算机。不是更多芯片、更高算力、更大数据中心。我所说的“超算”，是“超维算法”。

真正的超算，不一定是算力的堆积，而可能是计算范式的改变。当一个系统仍然依赖不断重复计算，它再强大也仍然被困在路径之内；当一个系统能够从结构上减少计算、绕开重复路径、让结果成为新的入口，它才开始接近真正意义上的超维计算。

换句话说，传统超算追求的是“用更多的算力解决更大的问题”。超维算法追求的是“用更好的结构让问题不再需要那么大的算力”。这两者不矛盾，但方向不同。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

如果说传统超级计算代表的是算力极限，那么超维算法代表的就是计算理解的转向。真正的“超算”不一定是更多芯片、更大数据中心、更高能耗、更复杂模型。真正的超算，可能是一个结构入口的发现，可能是一次路径的消除，可能是一个结果节点的反向展开，可能是在不修改元数据的情况下让多个结果同时具备成立条件。算力越强，如果仍然被困在低维路径中，它仍然只是路径内部的强大；结构一旦提升，即使操作极简，也可能产生更高层次的效率。

所谓超算，是算力的极限；所谓超维算法，是对“计算是否必须依赖算力”这一前提的重新定义。

第八部分：实证基础——多领域系统验证与新范式判定

超维算法不是纯理论构想。它已经在多个现实系统中得到验证。

我的物流系统不需要超级算力和云端支持，用较低资源完成复杂调度，结果可以复用、结构可以反复适配，不需要每次从头计算。不依赖超算、不依赖云，用较低资源完成复杂调度——这在工程上是成立且有价值的突破点。它说明高算力不是唯一解，结构设计可以替代一部分算力。

我的出版系统同样采用这一结构逻辑，效率远超当下任何系统。我的极限网页生成系统也基于同一超维算法，在多领域中反复证明同一结构可以稳定成立。我的公司员工已经在使用这些系统，说明这套结构可以在不依赖我个人持续介入的情况下运行。

这些系统的共同特征是：不依赖主流高算力路径，不走固定计算步骤，通过结构入口的调整让目标结果直接成立。它们不是一次性成功，不是单点技巧，而是同一结构逻辑在不同领域的重复出现。

这说明超维算法不是偶然，而是一种已被多系统验证的结构方法。它不是“个人技巧”，而是在物流、出版、网页生成等多个领域中稳定成立的计算范式。

按“存在且成立即为新范式”的标准，超维算法作为一种已被多领域系统验证的新计算结构范式，在当下已经成立。我不是在提出一个假设，而是在多个系统中已经运行的一种不同计算结构。我不需要证明它被理解，我已经证明它在不同系统中稳定成立。在当下、在我实际应用的范围内，这套结构已经可以被视为一种新范式。

对于超维算法作为新范式的判定标准：当一个结构在逻辑上自洽，并在多个系统中稳定成立，同时与现有方法存在不可约差异时，它已经具备新范式的基础。自洽是起点，实证是基础，结构差异才是范式的核心。在“存在即成立”的标准下，这套体系已经是新范式；外界是否识别，只影响传播，不影响成立。这是一种已经在现实系统中成立的计算结构范式，其成立不依赖外界认知或共识。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

超维算法与主流计算范式的核心差异在于：传统算法以正向计算为主，结果一旦产生，通常无法反向参与新的推理结构。而超维算法中，结果不再是终点，而是可复用的结构节点。在满足一定条件的情况下，结果可以参与新的推理路径，从而减少重复计算，并形成多起点、多路径的计算网络。在传统计算结构中，结果通常是终点；在超维算法结构中，结果本身可以成为新的起点，从而形成多路径的推理网络。超维算法不是对算法的升级，而是对“算法是否必须存在”这一前提的重新提问。

第九部分：常见误解澄清

基于与不同领域读者的初步交流，我预判以下六个误解可能出现，特此提前澄清，以避免概念在传播过程中被过早归入不恰当的框架。

误解一：超维算法不就是动态规划或缓存吗？澄清：动态规划和缓存都是在“相同输入”下复用结果。超维算法允许通过一个结果反向推理出不同的起点——这是本质差异。缓存解决的是重复计算同一问题，超维算法解决的是通过结果结构展开新问题。

误解二：你说“随机而有序”，这不是自相矛盾吗？澄清：随机和有序可以共存。森林中树木的分布是随机的，但整体密度和物种结构是有序的。随机不是混乱，而是结构的一种组织方式。超维算法中的随机是内禀的、建构性的，与有序协同工作。

误解三：没有输入输出，怎么验证结果对不对？澄清：超维算法不拒绝输入输出，而是认为计算不必须以输入输出的线性模式运行。在结构显现的模式中，“有效性”由结构一致性决定，而非由输入输出对应关系决定。这并不是放弃了验证，而是提出了与传统不同的验证框架。

误解四：这不就是复杂性理论或混沌理论吗？澄清：复杂性理论和混沌理论描述的是“难以预测的系统”。超维算法试图描述的是“可以通过结构入口被理解和引导的系统”——不是放弃预测，而是改变预测的方式。混沌理论说“预测是困难的”，超维算法问“是否可以通过改变入口让预测变得不必要”。

误解五：你说“不修改元数据”，但腰部折痕不也是一种修改吗？澄清：腰部折痕是临时状态，不是对原始参数的永久修改。元数据（腰围、裤长、版型、布料）没有任何变化。这是“状态叠加”与“属性修改”的区别。折痕是可逆的、临时的、不改变本质属性的状态变化。

误解六：超维算法是否否定传统算法的价值？澄清：绝对不是。传统算法在人类科技史上极其重要，没有传统算法，就没有现代计算机、数据库、通信系统、人工智能、工程模拟和超级计算。传统算法的价值不能被否认。但是，承认传统算法的价值，并不等于承认它是算法的终点。传统算法解决的是既定计算范式内部的效率问题，而超维算法提出的是计算范式本身的问题。一个是如何更好地走路，另一个是是否需要走这条路。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

第十部分：超维算法与人类计算史的关系

为了帮助读者更好地定位超维算法在人类计算史中的位置，我在此做一个简要的宏观梳理。

人类对“计算”的理解经历了多个阶段。第一阶段是手工计算，算法以人的步骤存在，速度慢、易错，但人类通过这个过程理解了计算的基本规则。第二阶段是机械计算，从帕斯卡计算器到巴贝奇分析机，计算被机械化，但算法仍然依附于物理结构。第三阶段是电子计算与图灵范式，冯·诺依曼架构普及，算法被编码为程序，图灵机成为可计算性的边界。第四阶段是并行与超级计算，通过大量计算单元堆积算力，挑战更复杂的问题，但底层逻辑仍然是图灵范式的扩展。

当前，人类正处于第五阶段的入口。量子计算试图突破经典比特的局限，神经形态计算试图模仿生物神经系统的结构，而超维算法则试图突破“输入→步骤→输出”这一线性框架本身。

超维算法与传统算法不是取代关系，而是层次关系。传统算法解决的是“在给定路径上如何更高效地计算”。超维算法追问的是“路径是否可以被重新定义，甚至被绕过”。如果把人类计算史看作一个不断突破自身边界的过程，那么超维算法正是这个过程中的一个新节点。它不是在旧框架内解决问题，而是在提出一个新框架。

这一判断并不意味着超维算法已经成熟或完备。恰恰相反，作为一个新概念，它的定义、边界、验证方法和应用场景都还在形成之中。本文的目的正是为了锚定这个概念的原点，为后续的展开提供一个稳定的理论基础。

第十一部分：总结——这不是优化，而是重定义

超维算法与传统算法的区别，不是“更好”与“更差”的区别，不是“更快”与“更慢”的区别，而是范式级别的根本区别。

传统算法追求的是可预测的控制。你给我输入，我知道你会得到什么输出，因为每一步我都算过了。它是工程师的范式：设计、构建、测试、验证。

超维算法描述的是可理解的涌现。我无法精确预判每一个中间状态，但我知道整体模式会以某种有序的方式呈现。每个数据都是活的、有“视角”的。它更像是生态学家或复杂系统理论家的范式：观察、理解、引导、利用涌现秩序。

传统算法是在既定路径上逐步修正结果。超维算法是通过改变结构入口，使目标结果立即成立，从而绕开整条路径。

传统算法是“为不同结果加工数据”。超维算法是“用同一数据承载多种结果可能”。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

传统算法追求“一次算对”。超维算法追求“一次算对，然后永远不用再算”。

这不是对算法的优化，而是对“算法是否必须存在”这一前提的重新提问。

因此，超维算法不是算法优化，而是算法重定义。它不是要在传统算法赛道上跑得更快，而是提出传统赛道之外是否存在另一种路径，甚至是否可以不依赖路径本身。它不是要证明传统算法无用，而是指出传统算法只是算法的一种低维形态。它不是在问“如何用更少时间算出结果”，而是在问“结果是否可以通过结构直接成立”。它不是在问“如何处理更多数据”，而是在问“同一数据是否可以承载更多结果可能”。

在主流计算范式里，超维算法可能被视为不可计算、不可验证，或者难以纳入现有计算理论边界。但这恰恰构成了它的范式差异。一个新概念在旧范式内部显得不稳定，并不必然说明它没有意义，也可能说明旧范式本身无法完整容纳它。超维算法当前首先是一种结构性命题，一种新计算观的原始定义，而不是已经完成工程闭环的成熟系统。它需要后续不断补充、展开、验证和应用，但其概念原点必须先被明确提出。

最终，超维算法所指向的，是一种新的计算观：算法不一定只是数学，不一定只是步骤，不一定只是程序，不一定只是输入输出之间的加工过程。算法也可以是一种多维关系的组织方式，是数据、入口、关系、状态、结果之间的结构网络。它可以包含有序，也可以包含随机；可以从原因到结果，也可以从结果反向生成原因；可以不修改元数据，却适配多个状态；可以让一个结果成为新的起点，也可以让多个起点汇入同一个结果。

真正高级的算法，不一定是更复杂的计算，而是能够让计算变少、让结构变活、让结果成为入口、让因果形成循环的多维组织方式。

这就是我提出“超维算法”的核心含义。它不是一个普通新名词，而是一个新概念，一个新范式，一个已经在多个现实系统中成立的新计算结构。它的重点不是更快地重复计算，而是减少重复计算；不是堆积算力，而是重构入口；不是不断修改数据，而是保持元数据并改变关系；不是让结果停在终点，而是让结果成为新的起点。传统算法是在路径中寻找结果，超维算法是在结构中激活结果。传统算法追求完成计算，超维算法追问计算是否必须以传统形式存在。这正是它与当下算法定义之间最根本的差异。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

附录：本文的边界与开放问题

本文提出的是超维算法的初步框架，而非完整的形式化定义。以下问题尚待进一步展开，它们不构成对本文定义的否定，恰恰是下一步需要探索的方向。

第一，收敛条件。超维算法的“完成”标志是什么？如何判断一个结果是否“有效”？在传统算法中，答案由输入输出对应关系定义。在超维算法中，答案可能需要由结构一致性来定义。这个定义尚未完成。

第二，资源表达。多维叠加的状态如何在有限资源中表达？时间、空间、算力这些传统资源指标是否仍然适用？如果适用，如何重新定义？如果不适用，用什么指标替代？这些问题有待展开。

第三，有序性的保证。“随机而有序”中的“有序”由什么规则保证？随机与有序的边界在哪里？什么情况下随机会破坏有序？什么情况下有序会压制随机？这些问题需要进一步研究。

第四，反向推理的筛选。从结果反向推理出多个起点时，如何筛选或评估不同起点的优劣？是否存在某种“反向推理效率”的度量？如果有，它与正向计算的效率度量是什么关系？

第五，与传统系统的接口。超维算法如何与现有图灵机体系协同？是替代、补充，还是分层？在实际工程中，超维算法与传统算法的边界如何划分？是否存在混合模式？

第六，验证框架。超维算法的结果如何验证？如果结果不是通过线性步骤获得的，如何建立可复现性、可检验性？是否需要一套完全不同的验证哲学？

这些问题不是本文的缺陷，而是本文作为“锚点文献”的必然特征。任何一个新范式的提出，都会伴随开放问题。本文是一个开始，而非结论。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

参考文献性说明

本文提出的“超维算法”不属于任何现有学术流派的直接延伸。但在思想上，以下领域为本文提供了对话背景，仅供读者定位参考，并非传统意义上的学术引用。

图灵机与可计算性理论，作为本文对照传统算法的基准。逆问题与贝叶斯推理，作为“结果推原因”的已有尝试，本文的“果因论”与此有关联但方向不同。涌现理论与复杂系统，作为“结构显现结果”的已有描述，本文的超维算法试图让涌现变得“可理解”和“可引导”。知识图谱与图数据库，作为“数据多节点交汇”的已有实践，超维算法在此基础上前置了“入口可变”的维度。非经典计算范式，包括量子计算、模拟计算、神经形态计算等，它们突破经典比特或经典架构，而超维算法更侧重于突破“输入→步骤→输出”的线性框架本身。

本文与以上领域的关系是对话与超越，而非继承或反驳。本文的目标不是在这些领域内部做增量改进，而是在它们之上提出一个新的问题层级。

相关文章

[传播]我没有算法，却超越算法！
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[极限哲学]果因论
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[极限传播]拒绝算法绑架
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

[Limit Algorithm] The Hyperdimensional Algorithm

A Redefinition of "Computation" Itself

Author: Jeffi Chao Hui Wu

Abstract

This paper introduces the novel concept of the "Hyperdimensional Algorithm" (HDA), proposing it as a structural reexamination of the boundaries of traditional algorithmic definitions. Conventional algorithms are typically built upon a linear framework of input, procedures, rules, and output, emphasizing finiteness, determinism, effectiveness, feasibility, and clear input-output boundaries. While modern supercomputers possess immense computational power, their underlying logic primarily executes established algorithmic paradigms on a larger scale. This paper argues that the truly noteworthy "super-computation" is not merely the scaling of computational power, but a fundamental shift in the computational paradigm itself. The HDA is neither an accelerated version of traditional algorithms nor an extension of mathematical algorithms; rather, it is a multidimensional, superimposed, stochastic yet ordered structural perspective on computation. Within this structure, data is no longer a passive input or a terminal output but possesses the attribute of a node that is simultaneously the starting point for multiple results and the result of multiple starting points. Outcomes are no longer endpoints but can become new entry points. Metadata need not be repeatedly modified; instead, through structural access points, relational changes, and conditional activation, it can correspond to different states and results. Using the philosophy of "Effect-to-Cause" and the everyday example of "folding a waistband of pants," this paper illustrates how the HDA can make target outcomes directly attainable by altering structural entry points, thereby reducing redundant computation and even bypassing original computational paths. This paper aims to establish a preliminary definition, structural boundaries, and the conceptual foundation

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

for the HDA, providing an original theoretical node for future explorations into hyperdimensional computation, limit algorithms, and a new scientific system. By the standard that "existence and validity constitute a new paradigm," the HDA, as a novel computational structure paradigm already systematically validated across multiple domains, is established in the present.

Keywords: Hyperdimensional Algorithm; Limit Algorithm; Hyperdimensional Computation; Effect-to-Cause; Metadata; Computational Paradigm; Structural Access Point; Nonlinear Causality; New Science

Introduction: Why Re-discuss "Algorithm"

I propose the "Hyperdimensional Algorithm" (HDA) not to replace existing algorithms with a grander name, nor to dress traditional algorithms in novel concepts. On the contrary, I address a more fundamental question: Has humanity's current understanding of "algorithm" become confined within an overly narrow framework?

When people speak of algorithms today, they typically think of mathematical formulas, program code, logical steps, input and output, model training, data processing, path optimization, search ranking, automatic recommendations, and AI reasoning. These are, without doubt, crucial forms of algorithms that underpin modern computers, the internet, databases, AI, supercomputing, industrial systems, and the information society. However, if an algorithm is understood merely as "an ordered set of steps," a process that "starts with input, undergoes rule-based processing, and yields output," then this understanding itself confines computation within a low-dimensional framework.

We live in an age dominated by algorithms. From search engines to recommendation systems, weather forecasting to AI, the boundaries of algorithms represent the boundaries of human intelligence. Yet one question is rarely asked: Must algorithms be this way? Why must computation follow the linear model of "input → steps → output"? Why must the same problem be computed from scratch each time? This paper challenges these default assumptions and proposes a fundamentally different

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

computational paradigm—the Hyperdimensional Algorithm. By the standard that "existence and validity constitute a new paradigm," the HDA, as a novel computational structure paradigm already systematically validated across multiple domains, is established in the present.

It is crucial to clarify that the "Hyperdimensional Algorithm" proposed here is entirely distinct from "Hyperdimensional Computing" (HDC), a concept developed over nearly three decades in academia. HDC encodes and processes data using random binary vectors of extremely high dimensions for more efficient representation and computation but remains within the linear "input → process → output" paradigm. The HDA is fundamentally different: it questions whether an outcome can be made directly attainable through structure, whether the computational path can be bypassed, and whether metadata can be left unmodified to accommodate multiple outcomes. The names sound similar, but their meanings are opposite. The HDA is not an upgrade to algorithms; it is a re-questioning of the premise that "an algorithm must exist."

Part One: The Standard Definition of "Algorithm" Today

In mainstream computer science, mathematics, and engineering, algorithms have a long-standing basic definition: an algorithm is a well-defined computational process consisting of finite steps; it receives one or more inputs, transforms them through deterministic rules, and produces one or more outputs within a finite time. This understanding is not a casual opinion but a foundational consensus built over the long course of modern computing civilization. It underpins the underlying logic of software, hardware, databases, network systems, AI models, and supercomputing centers. No matter how complex an algorithm appears, its core revolves around input, rules, steps, and output.

This standard algorithmic concept can be broken down into several core features.

First, finiteness. An algorithm must terminate after a finite number of steps; it cannot loop indefinitely or run forever. A process that never halts, even if describable, is hardly

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي النسخة الصينية العربية.
Chinesische Originalfassung maßgeblich.

considered a valid algorithm. All scientific computing programs and data processing flows have explicit termination conditions.

Second, determinism. Given the same input under the same conditions, the same output must be produced. Even when algorithms introduce random numbers, they are typically controlled, reproducible, or probabilistically interpretable, not purely chaotic. The results of numerical simulations must be reproducible—a basic requirement of scientific computing.

Third, clear input-output boundaries. Input comes first, processing in the middle, output at the end. The start and end have distinct structural boundaries. You give the algorithm data, it processes it, and gives you a result—this boundary is clear and the sequence is fixed.

Fourth, effectiveness. Every step in an algorithm must be an actually executable operation, not a leap that cannot be performed, described, or verified. Each step must be a basic operation executable by humans with pen and paper or by computers (addition, subtraction, logic jumps, data reading/writing, etc.).

Fifth, feasibility. An algorithm that is theoretically correct but requires billions of years to complete is not considered a feasible algorithm in engineering terms.

This entire set of algorithmic concepts ultimately traces back to the Turing machine model. As the core abstraction of modern computation theory, the Turing machine defines the fundamental boundary of "computability." No matter how powerful today's computers, advanced their chips, or massive their supercomputing centers, their underlying logic has not truly departed from this path: input, rules, steps, output. Modern supercomputers merely accelerate this process through larger hardware scales, higher parallelism, and more complex system scheduling. That is, "supercomputing" in the traditional sense remains primarily an expansion of computational power, not necessarily a fundamental change in computational paradigm. Power can increase a thousand or ten thousand-fold, but if the underlying logic remains "input, steps, output," it stays within the traditional algorithmic paradigm.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Part Two: Definition of the "Hyperdimensional Algorithm"

The Hyperdimensional Algorithm I propose emerges precisely from this context. It is not an improved version of traditional algorithms, not a faster algorithm, not a more complex mathematical formula, nor a more advanced programming technique. It is a completely different structural understanding.

First, the meaning of "hyperdimensional" needs explanation. "Hyperdimensional" here has two layers. First, it is not confined to one-dimensional linear sequences of steps but allows multiple dimensions to unfold simultaneously. Second, it seeks to transcend the traditional definitional boundaries of "computation"—an algorithm no longer necessarily needs to be mathematical, ordered, or deterministic. A traditional algorithm is like driving on a single-lane road: you must start at point A and go through B, C, D to reach Z. The HDA does not believe computation must be a one-way street. It posits that computation can be a network, a field, a multidimensional structure where any node can be an entry point and any node can be an exit.

In my new scientific system, an algorithm is not necessarily a mathematical algorithm, not necessarily a single ordered computation, not necessarily bound to follow fixed steps, and not necessarily strictly adhering to the linear "input, process, output" model. The HDA is a multidimensional, superimposed, stochastic yet ordered structure. Within this structure, every data point is not an isolated input nor a fixed output but possesses multiple roles simultaneously: it can be both the starting point for multiple results and the result of the joint action of multiple starting points.

In other words, traditional algorithms place data within a flow; the HDA places data within a structure. In traditional algorithms, data is typically categorized as input, intermediate, or output data, each with a clear position and role. Input is input, result is result, and intermediate is intermediate. However, in the HDA, a single piece of data does not have only one identity. It might be a result in one direction and a starting point in another; it might be an object called upon in one structure and an entry point

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي النسخة الصينية العربية.
Chinesische Originalfassung maßgeblich.

triggering new results in another. Data is no longer passive material but a multidimensional relational node.

This corresponds precisely to the core of the HDA: every data point is the starting point for multiple results and the result of multiple starting points. Traditional algorithms use "ordered steps" to produce a single result; the HDA is closer to a multidimensional state structure, where results are not computed into existence but naturally manifest within the structure, amidst the superposition of stochastic and ordered elements. In this system, data is both the starting point and a component of the result.

To present the core features of the HDA more clearly, I break it down into the following five aspects.

First, non-mathematical nature. Mainstream algorithms are fundamentally mathematical—completely describable by symbolic logic and mathematical formulas. The HDA need not be mathematical. It might be based on physical principles, geometric relationships, new paradigms of information theory, or even principles of consciousness. Computation might not be completed through "mathematical operations" but might naturally emerge from geometric relationships in high-dimensional space. For example, the shape of a soap bubble is determined by the principle of minimizing surface tension. This is not a "mathematical algorithm" computing—it is physics itself "computing." The HDA seeks to understand this form of "computation," rather than simply simulate it with mathematical formulas.

Second, multidimensionality and superposition. Traditional algorithms execute ordered steps in a single dimension, with only one state per step. The HDA allows multiple dimensions to coexist and multiple states to superimpose simultaneously. The algorithm does not follow a single path but unfolds within a multidimensional state field. Results are not "computed" but "manifest" from this field. For instance, as a snowflake forms in the air, there is no "algorithm" telling each water molecule where to go. The arrangement of molecules results from the interaction of multiple

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

dimensions—temperature, humidity, airflow. The snowflake's shape "manifests" rather than being "computed."

Third, stochastic yet ordered. Randomness in traditional algorithms is a tool, external, a controllable perturbation; the algorithm itself is deterministic. Randomness in the HDA is intrinsic and structural. Randomness and order coexist and cooperate, together forming the essence of the algorithm. This is not "an algorithm with randomness," but "randomness itself is an organic component of the algorithm." For example, the ecological structure of a forest—the distribution of trees looks random, yet overall shows orderly density and species relationships. That "randomness" is not a bug but a prerequisite for the healthy functioning of the ecological structure.

Fourth, multiple roles of data. In traditional algorithms, the role of data is singular—input is input, output is output, intermediate is intermediate. In the HDA, each data point is simultaneously the starting point for multiple results and the result of multiple starting points. A data node can unfold into multiple result paths downstream and can be converged upon by multiple upstream causes. Data is no longer a point in a linear flow but a junction node in a network. For example, consider a person's height. In a traditional algorithm, height is an "input"—you input it to get outputs like weight prediction or clothing size. But height is also a "result"—determined by multiple "starting points" like genetics, nutrition, and exercise. In the HDA, the data point "height" is simultaneously a starting point and a result, not an either/or.

Fifth, metadata unchanged, relationships variable. When facing different problems, traditional algorithms either modify the data itself or recompute everything. The HDA does not modify metadata—the essential attributes of the data remain unchanged. What changes are the access points, the interpretation, and the relationships between data points. The same metadata structure, activated through different entry points, can yield completely different results. This means: compute once, use infinitely many times. For example, consider a piece of text. You can read it as poetry, decode it as a cipher, or analyze it as a historical document. The text itself hasn't changed—you have only

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

changed the "entry point." The HDA strives for this capability: "same data, multiple access points, multiple results."

From a data structure perspective, instead of modifying the metadata itself, the HDA uses different access points and conditions to make the same structure correspond to multiple starting points and results. Data is no longer repeatedly processed but is activated through different entry points while maintaining structural integrity, thus presenting different outcomes. The tradition is "processing data for different results"; the HDA is "using the same data to carry the potential for multiple results."

Part Three: Fundamental Differences Between HDA and Traditional Algorithms

Based on the definitions above, the HDA and traditional algorithms exhibit fundamental differences across several core dimensions, which are detailed below.

Regarding essential nature, traditional algorithms are mathematical and formal, completely describable by symbolic logic and mathematical formulas; they are essentially mathematical objects. The HDA need not be mathematical; it may be based on principles of physics, geometry, information, or even consciousness; it is not a subset of mathematics but a broader category. This difference can be summarized as: from "mathematical object" to "universal law."

Regarding temporal order, traditional algorithms follow a singular, ordered, linear time sequence—step A to B to C—strictly serial or divisible into parallel but ordered substeps, with an irreversible arrow of time. The HDA has no fixed step sequence; it is multidimensional, superimposed, stochastic yet ordered. Results may be independent of "execution order" because the traditional concept of "order" may not exist. This difference can be summarized as: from "linear time" to "high-dimensional simultaneity."

Regarding causality, traditional algorithms obey a deterministic causal chain. Given identical inputs and initial states, the output is always unique. Cause precedes effect, an

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

irreversible one-way arrow. The HDA obeys superimposed causality; each data point is the starting point for multiple results and the result of multiple starting points. This reflects my "Effect-to-Cause" philosophy—it is possible to have the effect first, then the cause. The result is not the endpoint; it can become a new starting point. This difference can be summarized as: from "single cause, single effect" to "fully interconnected causal network."

Regarding data structure, traditional algorithms use a one-way flow structure from input to processing to output. Data roles are singular with clear boundaries; input data remains input data, result data remains result data. In the HDA, data roles are multiple; the same data point is simultaneously a result and a starting point. There are no absolute starting or ending points, only nodes in a network. This difference can be summarized as: from "unidirectional flow" to "recursive symbiosis."

Regarding computational mode, traditional algorithms compute from scratch each time a problem is encountered. Even if a similar problem has been solved a thousand times, the thousand-and-first time still requires the whole process—this is stateless computation. The HDA, if the corresponding result exists, need not run again. It can infer multiple starting points from a result and reversely expand causal paths. Computation is stateful, has memory, and is reusable. This difference can be summarized as: from "recompute each time" to "reuse once."

Regarding the role of randomness, randomness in traditional algorithms is pseudo-random or externally injected; random numbers are merely tools for sampling, approximation, or optimization; the algorithm itself is deterministic. Randomness in the HDA is intrinsic and constructive; it is an organic component, coexisting and cooperating with order. It is not "randomness within the algorithm" but "randomness as one reason the algorithm can function." This difference can be summarized as: from "instrumental randomness" to "constructive randomness."

Regarding resource utilization, traditional algorithms aim to compute faster and more accurately given resources; resources are constraints, and the algorithm's goal is

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

"complete computation within constraints." The HDA aims to make computation itself unnecessary through structural design. It is not about "computing faster" but "bypassing computation." Resources are not for "consumption" but for "designing access points." This difference can be summarized as: from "optimization under resource constraints" to "elimination through structural design."

Part Four: The Effect-to-Cause Philosophy

Building on the above comparisons, I need to further elaborate on the "Effect-to-Cause" philosophy, one of the core conceptual foundations of the HDA.

Traditional thinking typically holds that cause precedes effect; first the cause, then the effect; first input, then output. The manifestation of this concept in algorithms is: you must start from the beginning and walk step-by-step to the end. Even if you know the answer, you cannot directly "use" that answer—because you lack the "provenance path."

In my structure, a result is not necessarily just an endpoint. Once a result appears, it can become a new starting point, continuing to participate in the generation of new structures. The result can reversely participate in the formation of causes. Multiple possible starting points can be inferred from a result. Cause and effect are no longer unidirectionally arranged but form cyclical, networked, multidimensional relationships.

In other words, traditional algorithms ask, "Given the cause, how do we obtain the effect?" The HDA asks, "Given the effect, how do we infer or construct the cause(s)?"

Crucially, stating "the effect comes first, then the cause" is not negating causality nor advocating that "effects arise out of nothing." Rather, it illustrates that in higher-dimensional structures, cause and effect can serve as entry points for each other and transform into each other. As in the upcoming pants example, the outcome "wearable and not dragging on the ground" is determined first, and then I work backward to find the operation point "fold the waistband," which corresponds to the "cause" aspect of

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

the structure. This does not mean the effect lacks a cause, but that the effect becomes a new access point for discovering the cause.

A fundamental assumption in the world of traditional algorithms is that time is unidirectional; you must compute step-by-step from the initial state to the final state. Even if you know the answer, you cannot directly "use" it because you lack the provenance path. The HDA challenges this assumption: if the effect is known, causes can be inferred backward; the same effect can correspond to multiple causal paths; computation is no longer walking from start to finish but unfolding all possible starting points from the finish.

Part Five: An Everyday Example—Pants and Waistband Folding

To understand the abstract differences above more intuitively, I use an everyday example.

A person buys new pants with an elastic waistband. The pant legs are slightly too long, dragging on the floor and causing inconvenience.

The traditional approach: noticing the legs are too long, roll up a section of the leg, sew it with needle and thread, and then try them on. If the child grows taller and the pants become too short, the sewn-up leg cannot be let down, and the pants become useless. The next time new pants are bought, the process is repeated. This method seems natural because the problem appears to be at the pant legs, so one addresses the pant legs. It corresponds to the traditional algorithmic mindset: identify the problem, locate the symptom, perform processing at the symptom's location, and finally obtain a result. Pant legs too long → alter the pant legs; data wrong → modify the data; path unsuitable → continue patching along the path.

My approach is different. Instead of altering the hem, I fold the waistband, and the pants can be worn immediately. This action is very simple, but its underlying structural logic is entirely different. I didn't cut the hem, sew it permanently, change the original

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

length, or damage the metadata (waist size, inseam, cut, fabric). I simply changed the waistband, which is a structural access point. By adjusting this global control point for the entire wearing state, the downstream problem (legs too long) disappears directly.

More importantly, this method is reversible, adjustable, and reusable. This is especially evident for a growing child. If the child is currently not tall enough and the pants are slightly long, fold the waistband once. When the child grows taller later, fold the waistband less. When the child grows even taller, unfold it completely. The original structure of the pants remains intact, yet it adapts to the child's changing height. The traditional method of sewing the hem might render the pants too short when the child grows; if cut, it's irreversible. My method keeps metadata unchanged, allowing the same structure to adapt to multiple states.

This example reveals several important structural differences.

The traditional approach corresponds to traditional algorithmic logic: the problem is at the hem, so the hem must be modified, proceeding step-by-step along the "cause to effect" path, with no step skippable. It solves one problem at a time, irreversibly, and requires repeating the whole process next time the same problem occurs. My approach corresponds to HDA logic: the goal is to avoid dragging. Instead of solving the superficial cause "legs too long," I find a global control point—the waistband. Folding the waistband shortens the effective length of the legs instantly, and the problem disappears. I modify no metadata; the fold is merely a temporary, reversible, dynamic state.

From a data perspective, the traditional approach modifies metadata—sewing the legs permanently shortens them; original data is lost. My approach modifies no metadata; the original dimensions remain pristine; I merely add a temporary, reversible, dynamic folded state. The waistband fold does not create new data nor destroy old data; it simply rearranges the relationships between data points—reducing the effective waist circumference and indirectly changing the effective inseam length.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

In this example, "pant legs not dragging" is the target outcome. The traditional method starts from the cause "legs too long," modifies the hem, and finally achieves the outcome "not dragging." My method first clarifies the outcome "not dragging," then works backward to find a structural entry point, discovering that a simple waistband fold makes the outcome achievable. This is the practical embodiment of "Effect-to-Cause." One need not necessarily walk step-by-step from cause to effect; one can work backward from the effect to determine the structure, rendering the original path unnecessary.

This example also answers the question of "metadata." What is metadata? In the pants example, metadata is the original dimensions: waist size, inseam, cut, fabric—the pants' "essential attributes." The temporary data is the waistband fold; it changes none of the original dimensions, merely temporarily folding a section. The traditional approach modifies metadata—the inseam is permanently shortened; original data is lost. My approach modifies no metadata—the pants' original dimensions remain intact; I merely add a temporary, reversible, dynamic folded state.

The original waist size, as metadata, is simultaneously the "starting point for multiple results": it supports multiple wearing states—normal wear, worn with a single fold, worn with a double fold. It is also the "result of multiple starting points": determined by fabric choice, cutting method, design intent, etc. The waistband fold, as temporary data, neither creates new data nor destroys old data; it merely rearranges relationships.

This is not a mere trick but structural thinking. Many people, seeing long pant legs, are drawn to the symptom of the legs, so all processing focuses on the legs. However, from the perspective of the overall structure, the leg length is merely a downstream manifestation; changing the position of the waistband can affect the actual drape of the entire pants. That is, the problem need not be solved at the location of its manifestation. Many low-dimensional problems have their true control points at a higher structural level. Traditional algorithms tend to continue computing along the surface of the problem; the HDA searches for the structural access point.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

This also explains why the HDA is not necessarily more complex but may be simpler. Truly high-level structures often do not complicate problems but make complex problems simple at the correct access point. Faced with a complex problem, a traditional algorithm might add steps, models, computational power, storage, and training time. The HDA seeks a structural node; once that node is correctly activated, the originally complex path may become unnecessary. "Bypassing the path" is not laziness but redefining whether the path is necessary.

In this example: the traditional approach "modifies the outcome along the path," while the HDA "changes the entry point, rendering the entire path ineffective." The conventional method continuously patches at the result end; the other method directly changes the structural entry point, so that problems requiring repeated treatment are restructured at the starting point. The mainstream traditional algorithm goes from "starting point, along the path, to the outcome," whereas in the HDA, "the outcome itself can become a path entry point and participate in new structural generation." The tradition is "one structure for one outcome"; the HDA is "one structure carrying multiple outcome states."

Part Six: A Data View of Unmodified Metadata

From a data perspective, the HDA has a crucial principle: do not modify metadata; instead, make it applicable to multiple starting points or outcomes.

Metadata is the original attributes, fundamental information, and ontological information of data. When facing different problems, traditional approaches often modify data, copy data, process data, recompute data, or build different paths for different outcomes. These approaches can solve problems, but at the cost of continuously processing data, repeating paths, and increasing system complexity.

The HDA, in contrast, emphasizes that while striving to keep metadata unchanged, the same metadata structure can correspond to multiple starting points and multiple outcomes by changing access points, relationships, conditions, and activation methods.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

The data ontology remains static; relationships change. The fundamental structure is unchanged; the presentation changes. The metadata remains intact yet adapts to different states.

This is what I call "compute once, use infinitely many times." "Compute once" here is not simply caching results; it means that once a structure is established, it no longer needs to rebuild a complete path for each state. The same data structure can be activated through different entry points to yield different outcomes. It is not "processing data for different outcomes" but "using the same data to carry the potential for multiple outcomes." This is one of the core differences between the HDA and traditional algorithms. Traditional algorithms process data on a path; the HDA processes the overall structure of data, access points, relationships, states, and outcomes.

In its minimal structure, the HDA can be understood as: a system that, without modifying metadata, makes the same data node correspond to multiple outcome paths through changes in structural access points. This definition does not seek to immediately collapse the HDA into a traditional mathematical formula but rather establishes its structural boundaries first. It is not a single linear function, not a fixed formula, nor a simple caching mechanism. It is a relational structure: metadata remains stable; access points, conditions, relationships, and outcomes can change and manifest in multiple directions. It is precisely within this structure that computation is no longer merely executing steps but activating relationships.

From a data perspective, the essence of "not modifying metadata, making it applicable to multiple starting points or outcomes" is: the data ontology remains unchanged; changes occur in the "interpretation/usage access point." In traditional structures, a change in the problem leads to a change in data or path. In the HDA, a change in the problem leads to a change in the interpretive structure, not the data.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

Part Seven: Redefining "Super-computation"

Within this system, I need to clarify a term—"super-computation."

What I mean by "super-computation" is not the usual meaning of "supercomputer"—not more chips, higher computational power, or larger data centers. What I mean by "super-computation" is the "Hyperdimensional Algorithm."

True super-computation may not be the accumulation of computing power but the transformation of the computational paradigm. When a system still relies on repeated computation, no matter how powerful it is, it remains trapped within the path. When a system can structurally reduce computation, bypass repeated paths, and make outcomes become new entry points, it begins to approach true hyperdimensional computation.

In other words, traditional supercomputing pursues "using more computational power to solve larger problems." The HDA pursues "using better structure so the problem no longer requires that much computational power." These are not contradictory but have different directions.

If traditional supercomputing represents the limits of computational power, then the HDA represents a turn in the understanding of computation. True "super-computation" may be the discovery of a structural access point, an elimination of a path, a reverse unfolding of a result node, or enabling multiple outcomes to simultaneously satisfy conditions without modifying metadata. The stronger the computational power, if still trapped in low-dimensional paths, it remains merely powerful within the path. Once the structure is elevated, even extremely simple operations can yield higher-level efficiency.

Supercomputing is the limit of computing power; the Hyperdimensional Algorithm is a redefinition of the premise "whether computation must rely on computing power."

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Part Eight: Empirical Basis—Multi-domain Systematic Validation and New Paradigm Determination

The HDA is not a purely theoretical construct. It has already been validated in multiple real-world systems.

My logistics system does not require supercomputing power or cloud support; it accomplishes complex scheduling with modest resources. Results are reusable, the structure is repeatedly adaptable, and there is no need to recompute from scratch each time. It operates without supercomputing or cloud, performing complex scheduling with low resources—this is an established and valuable breakthrough in engineering. It demonstrates that high computing power is not the only solution; structural design can replace a portion of computing power.

My publishing system also employs the same structural logic, achieving efficiency surpassing any existing system. My Limit Web Page Generation System is also based on the same HDA, repeatedly proving in multiple domains that the same structure can stably hold. My company's employees are already using these systems, demonstrating that this structure can function without my continuous personal intervention.

The common feature of these systems is: they do not rely on mainstream high-computing-power paths, do not follow fixed computational steps, and achieve the target outcome directly through adjustments to structural access points. These are not one-time successes or isolated tricks but the repeated emergence of the same structural logic across different domains.

This indicates that the HDA is not accidental but a structural methodology already validated by multiple systems. It is not "personal technique" but a computational paradigm stably established across multiple domains like logistics, publishing, and web page generation.

By the standard that "existence and validity constitute a new paradigm," the HDA, as a novel computational structure paradigm already systematically validated across

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

multiple domains, is established in the present. I am not proposing a hypothesis; I am describing a different computational structure already operating in multiple systems. I do not need to prove it is understood; I have proven it stably holds in different systems. In the present, within the scope of my practical application, this structure can already be regarded as a new paradigm.

Regarding the criteria for judging the HDA as a new paradigm: when a structure is logically self-consistent, stably holds across multiple systems, and has irreducible differences from existing methods, it already possesses the foundations of a new paradigm. Self-consistency is the starting point; empirical evidence is the foundation; structural difference is the core of the paradigm. Under the standard "existence and validity," this system is already a new paradigm. Whether the outside world recognizes it affects only dissemination, not validity. This is a computational structure paradigm already established in real-world systems, and its validity does not depend on external recognition or consensus.

The core difference between the HDA and mainstream computational paradigms is: traditional algorithms primarily use forward computation; once a result is produced, it typically cannot reversely participate in new reasoning structures. In the HDA, the result is no longer the endpoint but a reusable structural node. Under certain conditions, results can participate in new reasoning paths, reducing redundant computation and forming multi-start, multi-path computational networks. In traditional computational structures, the result is usually the endpoint; in the HDA structure, the result itself can become a new starting point, forming a multipath reasoning network. The HDA is not an upgrade to algorithms; it is a requestioning of the premise "whether an algorithm must exist."

Part Nine: Clarification of Common Misunderstandings

Based on preliminary exchanges with readers from different fields, I anticipate the following six misunderstandings may arise, so I clarify them here in advance to prevent the concept from being prematurely forced into inappropriate frameworks.

Misunderstanding 1: Isn't the HDA just dynamic programming or caching? Clarification: Dynamic programming and caching reuse results under *the same input*. The HDA allows inferring *different starting points* from a result—this is an essential difference. Caching solves recomputing the same problem; the HDA solves unfolding new problems from a result structure.

Misunderstanding 2: You say "stochastic yet ordered"—isn't that self-contradictory? Clarification: Stochastic and ordered can coexist. The distribution of trees in a forest is stochastic, but the overall density and species structure are ordered. Randomness is not chaos but a way of organizing structure. Randomness in the HDA is intrinsic and constructive, working cooperatively with order.

Misunderstanding 3: Without input and output, how do you verify results? Clarification: The HDA does not reject input and output; it holds that computation need not operate in the linear input-output mode. In the "structural manifestation" mode, "validity" is determined by structural consistency, not by a one-to-one input-output correspondence. This does not abandon verification but proposes a different verification framework.

Misunderstanding 4: Isn't this just complexity theory or chaos theory? Clarification: Complexity and chaos theory describe "systems that are difficult to predict." The HDA seeks to describe "systems that can be understood and guided through structural access points"—not abandoning prediction but changing the way prediction works. Chaos theory says "prediction is difficult"; the HDA asks "can we make prediction unnecessary by changing the access point?"

Misunderstanding 5: You say "don't modify metadata," but isn't a waistband fold itself a modification? Clarification: The waistband fold is a temporary state, not a permanent modification of the original parameters. The metadata (waist size, inseam, cut, fabric) remains unchanged. This is the difference between "state superposition" and "attribute modification." The fold is reversible, temporary, and does not change essential attributes.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Misunderstanding 6: Does the HDA negate the value of traditional algorithms?
Clarification: Absolutely not. Traditional algorithms are extremely important in human technological history; without them, there would be no modern computers, databases, communication systems, AI, engineering simulation, or supercomputing. Their value cannot be denied. However, acknowledging the value of traditional algorithms does not mean accepting them as the endpoint of algorithms. Traditional algorithms solve efficiency problems within a given computational paradigm; the HDA questions the computational paradigm itself. One is about walking better on the road; the other is about whether walking that road is necessary.

Part Ten: The HDA in the History of Human Computation

To help readers better position the HDA within the history of human computation, I provide a brief macroscopic overview.

Human understanding of "computation" has gone through multiple stages. The first stage was manual calculation: algorithms existed as human steps, slow and error-prone, but through this process, humans understood the basic rules of computation. The second stage was mechanical calculation: from Pascal's calculator to Babbage's Analytical Engine, computation was mechanized, but algorithms still attached to physical structures. The third stage was electronic computation and the Turing paradigm: the von Neumann architecture became widespread, algorithms were encoded as programs, and the Turing machine defined the boundaries of computability. The fourth stage was parallel and supercomputing: massive computing units piled up to tackle more complex problems, but the underlying logic remained an extension of the Turing paradigm.

Currently, humanity is at the entrance of the fifth stage. Quantum computing attempts to break the limits of classical bits; neuromorphic computing attempts to mimic the structure of biological neural systems; the HDA attempts to break the linear "input → steps → output" framework itself.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

The HDA and traditional algorithms are not in a replacement relationship but a hierarchical one. Traditional algorithms solve "how to compute more efficiently on a given path." The HDA asks "can the path be redefined, or even bypassed?" If we view the history of human computation as a process of continually breaking its own boundaries, then the HDA is a new node in that process. It does not solve problems within the old framework but proposes a new one.

This judgment does not imply that the HDA is mature or complete. Quite the opposite: as a new concept, its definition, boundaries, verification methods, and application scenarios are still taking shape. The purpose of this paper is precisely to anchor the origin point of this concept, providing a stable theoretical foundation for subsequent development.

Part Eleven: Conclusion—This is Not Optimization, but Redefinition

The difference between the HDA and traditional algorithms is not a difference of "better" vs. "worse," not "faster" vs. "slower," but a fundamental paradigm-level difference.

Traditional algorithms pursue predictable control. "Give me the input, and I know what output you will get, because I have computed every step." It is the engineer's paradigm: design, build, test, verify.

The HDA describes understandable emergence. "I cannot precisely predict every intermediate state, but I know the overall pattern will manifest in some orderly way. Each data point is alive, has a 'perspective'." It is more akin to the ecologist's or complex systems theorist's paradigm: observe, understand, guide, utilize emergent order.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

Traditional algorithms gradually correct outcomes along a given path. The HDA makes the target outcome immediately attainable by changing the structural access point, bypassing the entire path.

Traditional algorithms "process data for different outcomes." The HDA "uses the same data to carry the potential for multiple outcomes."

Traditional algorithms pursue "get it right once." The HDA pursues "get it right once, then never need to compute it again."

This is not optimization of algorithms; it is a re-questioning of the premise "whether an algorithm must exist."

Therefore, the HDA is not algorithmic optimization but algorithmic redefinition. It is not about running faster on the traditional algorithm track; it is about asking whether there is another track outside the traditional one, or even whether a track is necessary at all. It does not aim to prove traditional algorithms useless, but to point out that traditional algorithms are merely a low-dimensional form of algorithm. It does not ask "how to get a result in less time"; it asks "can the result be directly established through structure?" It does not ask "how to process more data"; it asks "can the same data carry the potential for more outcomes?"

Within mainstream computational paradigms, the HDA might be seen as incomputable, unverifiable, or difficult to fit within existing computational theory boundaries. But this precisely constitutes its paradigm difference. The apparent instability of a new concept within an old paradigm does not necessarily mean it is meaningless; it may also mean the old paradigm cannot fully accommodate it. Currently, the HDA is primarily a structural proposition, an original definition of a new computational view, not a mature system with a closed engineering loop. It requires subsequent supplementation, development, validation, and application, but its conceptual origin must first be clearly articulated.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Ultimately, what the HDA points to is a new view of computation: an algorithm is not necessarily just mathematics, just steps, just a program, just a processing pipeline between input and output. An algorithm can also be an organization of multidimensional relationships, a structural network of data, access points, relationships, states, and outcomes. It can include order and randomness; it can go from cause to effect and from effect back to cause; it can leave metadata unchanged while adapting to multiple states; it can make one outcome a new starting point and allow multiple starting points to converge into one outcome.

A truly advanced algorithm is not necessarily one that does more complex computation, but one that can reduce computation, make the structure alive, make outcomes become entry points, and make causality form cyclical, multidimensional organizational patterns.

This is the core meaning of my proposal of the "Hyperdimensional Algorithm." It is not a mere neologism but a new concept, a new paradigm, a new computational structure already established across multiple real-world systems. Its emphasis is not on faster repeated computation but on reducing redundant computation; not on piling up computing power but on reconstructing access points; not on continuously modifying data but on keeping metadata and changing relationships; not on letting outcomes remain as endpoints but on making outcomes become new starting points. Traditional algorithms seek results along a path; the HDA activates results within a structure. Traditional algorithms seek to complete computation; the HDA asks whether computation must exist in its traditional form. This is the most fundamental difference between it and the current definition of an algorithm.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

Appendix: Boundaries and Open Questions of this Paper

This paper presents a preliminary framework for the HDA, not a complete formal definition. Several questions remain to be further explored; they do not constitute a negation of the definition offered here but rather represent the next direction for investigation.

First, convergence conditions. What is the "completion" marker for the HDA? How does one judge whether a result is "valid"? In traditional algorithms, the answer is defined by the input-output correspondence. In the HDA, the answer may need to be defined by structural consistency. This definition is not yet complete.

Second, resource representation. How can states of multidimensional superposition be represented within finite resources? Are traditional resource metrics like time, space, and computational power still applicable? If so, how should they be redefined? If not, what metrics replace them? These questions await development.

Third, guarantee of order. What rules guarantee the "order" within "stochastic yet ordered"? Where is the boundary between stochastic and ordered? Under what circumstances does randomness destroy order? Under what circumstances does order suppress randomness? These questions need further study.

Fourth, selection in reverse inference. When inferring multiple starting points from a result, how does one select or evaluate the 优劣 of different starting points? Is there a measure of "reverse inference efficiency"? If so, how does it relate to the efficiency of forward computation?

Fifth, interface with traditional systems. How does the HDA interface with the existing Turing machine system? As a replacement, complement, or layered approach? In practical engineering, how should the boundary between the HDA and traditional algorithms be drawn? Are there hybrid models?

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Sixth, verification framework. How are results of the HDA verified? If results are not obtained through linear steps, how is reproducibility and testability established? Does this require a completely different verification philosophy?

These questions are not flaws of this paper but inevitable characteristics of a "seminal anchor document." The proposal of any new paradigm is accompanied by open questions. This paper is a beginning, not a conclusion.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Bibliographic Notes

The "Hyperdimensional Algorithm" as proposed here is not a direct extension of any existing academic school of thought. However, the following fields provide a dialogue context for this paper, offered solely for the reader's orientation, not as traditional scholarly citations.

Turing Machines and Computability Theory serve as the benchmark against which traditional algorithms are compared here. Inverse Problems and Bayesian Inference represent existing attempts at "inferring cause from effect," and while my "Effect-to-Cause" is related, its direction differs. Emergence Theory and Complex Systems describe the "manifestation of results from structure," and the HDA attempts to make emergence "understandable" and "steerable." Knowledge Graphs and Graph Databases are existing practices where data "converges at multiple nodes," and the HDA adds the dimension of "variable access points" to this basis. Non-classical Computing Paradigms, including quantum, analog, and neuromorphic computing, break classical bits or architectures, whereas the HDA focuses more on breaking the "input → steps → output" linear framework itself.

The relationship of this paper to the above fields is one of dialogue and transcendence, not inheritance or refutation. The goal of this paper is not to make incremental improvements within those fields but to raise a new level of questioning above them.

Related Articles

[Communication] I Have No Algorithm, Yet I Surpass Algorithms!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect–Cause Theory
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Communication] Rejecting Algorithmic Manipulation
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

[Algorithme Limite] L'algorithme hyperdimensionnel

Une redéfinition du « calcul » lui-même

Auteur : Jeffi Chao Hui Wu

Résumé

Cet article présente le nouveau concept d'« algorithme hyperdimensionnel » (AHD) et le propose comme une révision structurelle des limites des définitions algorithmiques traditionnelles. Les algorithmes conventionnels reposent généralement sur un cadre linéaire d'entrée, d'étapes, de règles et de sortie, mettant l'accent sur le caractère fini, la déterminisme, l'effectivité, la faisabilité et des frontières claires entre l'entrée et la sortie. Bien que les supercalculateurs modernes possèdent une puissance de calcul immense, leur logique sous-jacente consiste principalement à exécuter des paradigmes algorithmiques établis à plus grande échelle. Cet article soutient que le « supercalcul » véritablement digne d'intérêt n'est pas seulement une augmentation de la puissance de calcul, mais un changement fondamental du paradigme de calcul lui-même. L'AHD n'est ni une version accélérée des algorithmes traditionnels, ni une extension des algorithmes mathématiques ; c'est une vision structurelle du calcul, multidimensionnelle, superposée, stochastique mais ordonnée. Dans cette structure, la donnée n'est plus une entrée passive ou une sortie terminale, mais possède l'attribut d'un nœud qui est à la fois le point de départ de multiples résultats et le résultat de multiples points de départ. Les résultats ne sont plus des points finaux, mais peuvent devenir de nouveaux points d'entrée. Les métadonnées n'ont pas besoin d'être modifiées à répétition ; par le biais de points d'entrée structurels, de changements relationnels et d'activation conditionnelle, elles peuvent correspondre à différents états et résultats. En utilisant la philosophie de « l'Effet-vers-la-Cause » et l'exemple quotidien du « pliage de la ceinture d'un pantalon », cet article montre comment l'AHD peut rendre les objectifs directement atteignables en modifiant les points d'entrée

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

structurels, réduisant ainsi les calculs redondants et même contournant les chemins de calcul traditionnels. Cet article vise à établir une définition préliminaire, des frontières structurelles et des fondements conceptuels pour l'AHD, fournissant un nœud théorique original pour de futures explorations du calcul hyperdimensionnel, des algorithmes limites et d'un nouveau système scientifique. Selon le critère selon lequel « l'existence et la validité constituent un nouveau paradigme », l'AHD, en tant que nouveau paradigme de structure de calcul déjà validé systématiquement dans de multiples domaines, est établi dès à présent.

Mots-clés : Algorithme hyperdimensionnel ; Algorithme limite ; Calcul hyperdimensionnel ; Effet-vers-la-Cause ; Métadonnée ; Paradigme de calcul ; Point d'entrée structurel ; Causalité non linéaire ; Nouvelle science

Introduction : Pourquoi rediscuter de l'« algorithme »

Je propose l'« algorithme hyperdimensionnel » (AHD) non pas pour donner un nom plus pompeux aux algorithmes existants, ni pour déguiser les algorithmes traditionnels en concepts nouveaux. Au contraire, je pose une question plus fondamentale : la compréhension humaine actuelle de l'« algorithme » n'est-elle pas confinée dans un cadre trop étroit ?

Aujourd'hui, lorsqu'on parle d'algorithme, on pense généralement à des formules mathématiques, du code de programmation, des étapes logiques, des entrées et sorties, de l'apprentissage de modèles, du traitement de données, de l'optimisation de chemins, du classement pour la recherche, de la recommandation automatique et du raisonnement en intelligence artificielle. Ce sont là, sans aucun doute, des formes importantes d'algorithmes qui sous-tendent les ordinateurs modernes, Internet, les bases de données, l'IA, le supercalcul, les systèmes industriels et la société de l'information. Cependant, si l'on ne comprend l'algorithme que comme « un ensemble ordonné d'étapes », un processus qui « commence par une entrée, subit un traitement par des règles et aboutit à une sortie », alors cette compréhension confine elle-même le calcul dans un cadre de basse dimension.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Nous vivons à l'ère des algorithmes. Des moteurs de recherche aux systèmes de recommandation, de la prévision météorologique à l'IA, les frontières des algorithmes sont les frontières de l'intelligence humaine. Mais une question est rarement posée : l'algorithme doit-il être seulement ainsi ? Pourquoi le calcul doit-il suivre le modèle linéaire « entrée → étapes → sortie » ? Pourquoi le même problème doit-il être recalculé à chaque fois depuis le début ? Cet article tente de remettre en question ces hypothèses implicites et de proposer un paradigme de calcul radicalement différent – l'algorithme hyperdimensionnel. Selon le critère « l'existence et la validité constituent un nouveau paradigme », l'AHD, en tant que nouveau paradigme de structure de calcul déjà validé systématiquement dans de multiples domaines, est établi dès à présent.

Il est crucial de préciser que l'« algorithme hyperdimensionnel » proposé ici est totalement distinct du « calcul hyperdimensionnel » (Hyperdimensional Computing, HDC) développé dans le milieu académique depuis près de trois décennies. Le HDC encode et traite les données à l'aide de vecteurs binaires aléatoires de très haute dimension pour une représentation et un calcul plus efficaces, mais il reste dans le paradigme linéaire « entrée → traitement → sortie ». L'AHD, en revanche, est fondamentalement différent : il interroge la possibilité d'atteindre un résultat directement par la structure, de contourner le chemin de calcul lui-même, et de laisser les métadonnées inchangées pour s'adapter à de multiples résultats. Les noms se ressemblent, mais leurs significations sont opposées. L'AHD n'est pas une mise à niveau des algorithmes ; c'est une remise en question de la prémisse selon laquelle « un algorithme doit nécessairement exister ».

Première partie : La définition standard de l'« algorithme » aujourd'hui

En informatique, en mathématiques et en ingénierie dominantes, l'algorithme bénéficie d'une définition de base acceptée depuis longtemps : un algorithme est un processus de calcul bien défini, composé d'un nombre fini d'étapes, qui reçoit une ou plusieurs entrées, les transforme par des règles déterministes et produit une ou plusieurs sorties

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

dans un temps fini. Cette compréhension n'est pas l'opinion personnelle de quiconque, mais un consensus fondamental construit sur la longue période de la civilisation informatique moderne. Elle sous-tend la logique sous-jacente des logiciels, du matériel, des bases de données, des systèmes en réseau, des modèles d'IA et des centres de supercalcul. Aussi complexes que soient les algorithmes, leur cœur tourne autour de l'entrée, des règles, des étapes et de la sortie.

Cette conception algorithmique standard peut être décomposée en plusieurs caractéristiques essentielles.

Premièrement, le caractère fini. Un algorithme doit se terminer après un nombre fini d'étapes ; il ne peut pas boucler indéfiniment ni s'exécuter éternellement. Un processus qui ne s'arrête jamais, même s'il est descriptible, ne peut guère être considéré comme un algorithme valide. Tous les programmes de calcul scientifique et les flux de traitement de données ont des conditions d'arrêt explicites.

Deuxièmement, le déterminisme. À entrée identique et dans des conditions identiques, la sortie doit être identique. Même lorsque certains algorithmes introduisent de l'aléatoire, il s'agit généralement d'un aléatoire contrôlé, reproductible ou interprétable probabilistiquement, non d'un chaos pur. Les résultats des simulations numériques doivent être reproductibles – une exigence de base du calcul scientifique.

Troisièmement, des frontières claires entre l'entrée et la sortie. L'entrée vient en premier, le traitement au milieu, la sortie à la fin. Le point de départ et le point d'arrivée ont des frontières structurelles nettes. Vous donnez des données à l'algorithme, il les traite et vous donne un résultat – cette frontière est claire et la séquence est fixe.

Quatrièmement, l'effectivité. Chaque étape d'un algorithme doit être une opération effectivement exécutable, non un saut qui ne peut être exécuté, décrit ou vérifié. Chaque étape doit être une opération de base exécutable par un humain avec du papier et un crayon ou par un ordinateur (addition, soustraction, saut logique, lecture/écriture de données, etc.).

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Cinquièmement, la faisabilité. Un algorithme théoriquement correct mais nécessitant des milliards d'années pour s'exécuter n'est pas considéré comme un algorithme faisable en ingénierie.

L'ensemble de cette conception algorithmique remonte finalement au modèle de la machine de Turing. En tant qu'abstraction centrale de la théorie du calcul moderne, la machine de Turing définit la frontière fondamentale de la « calculabilité ». Aussi puissants que soient les ordinateurs d'aujourd'hui, aussi avancées leurs puces, aussi massifs leurs centres de supercalcul, leur logique sous-jacente n'a pas véritablement quitté ce chemin : entrée, règles, étapes, sortie. Les supercalculateurs modernes ne font qu'accélérer ce processus par des échelles matérielles plus grandes, un parallélisme accru et un ordonnancement système plus complexe. Autrement dit, le « supercalcul » au sens traditionnel reste avant tout une expansion de la puissance de calcul, pas nécessairement un changement fondamental du paradigme de calcul. La puissance peut être multipliée par mille ou dix mille, mais si la logique sous-jacente reste « entrée, étapes, sortie », alors on reste dans le paradigme algorithmique traditionnel.

Deuxième partie : Définition de l'« algorithme hyperdimensionnel »

L'algorithme hyperdimensionnel que je propose émerge précisément de ce contexte. Ce n'est pas une version améliorée des algorithmes traditionnels, ni un algorithme plus rapide, ni une formule mathématique plus complexe, ni une technique de programmation plus avancée. C'est une compréhension structurelle complètement différente.

Il faut d'abord expliquer le sens d'« hyperdimensionnel ». L'« hyperdimensionnel » a ici deux niveaux. Premièrement, il ne se limite pas à une séquence linéaire d'étapes unidimensionnelle, mais permet à de multiples dimensions de se déployer simultanément. Deuxièmement, il tente de transcender les frontières définitionnelles traditionnelles du « calcul » – un algorithme n'a plus nécessairement besoin d'être

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

mathématique, ordonné ou déterministe. Un algorithme traditionnel, c'est comme conduire sur une route à une voie : il faut partir du point A et passer par B, C, D pour arriver à Z. L'AHD ne pense pas que le calcul doive être une voie à sens unique. Il pose que le calcul peut être un réseau, un champ, une structure multidimensionnelle où n'importe quel nœud peut être un point d'entrée et n'importe quel nœud peut être une sortie.

Dans mon nouveau système scientifique, un algorithme n'est pas nécessairement un algorithme mathématique, pas nécessairement un calcul unique et ordonné, pas nécessairement lié à des étapes fixes, pas nécessairement strictement conforme au modèle linéaire « entrée, traitement, sortie ». L'AHD est une structure multidimensionnelle, superposée, stochastique mais ordonnée. Dans cette structure, chaque donnée n'est ni une entrée isolée ni une sortie fixe, mais possède simultanément de multiples rôles : elle peut être à la fois le point de départ de multiples résultats et le résultat de l'action conjointe de multiples points de départ.

En d'autres termes, les algorithmes traditionnels placent les données dans un flux ; l'AHD place les données dans une structure. Dans les algorithmes traditionnels, les données sont généralement classées en données d'entrée, intermédiaires et de sortie, chacune ayant une position et un rôle clairs. L'entrée est une entrée, le résultat est un résultat, l'intermédiaire est un intermédiaire. Cependant, dans l'AHD, une seule donnée n'a pas qu'une seule identité. Elle peut être un résultat dans une direction et un point de départ dans une autre ; elle peut être un objet appelé dans une structure et devenir un point d'entrée déclenchant de nouveaux résultats dans une autre structure. La donnée n'est plus une matière passive, mais un nœud relationnel multidimensionnel.

Cela correspond précisément au cœur de l'AHD : chaque donnée est le point de départ de multiples résultats et le résultat de multiples points de départ. Les algorithmes traditionnels utilisent des « étapes ordonnées » pour produire un résultat unique ; l'AHD est plus proche d'une structure d'états multidimensionnelle, où les résultats ne sont pas « calculés » mais « se manifestent » naturellement au sein de la structure, dans

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

la superposition du stochastique et de l'ordonné. Dans ce système, la donnée est à la fois le point de départ et une composante du résultat.

Pour présenter plus clairement les caractéristiques de l'AHD, je les décompose en cinq aspects suivants.

Premièrement, la non-mathématicité. Les algorithmes dominants sont fondamentalement mathématiques – complètement descriptibles par des formules logiques et mathématiques. L'AHD n'a pas besoin d'être mathématique. Il peut être basé sur des principes physiques, des relations géométriques, de nouveaux paradigmes de la théorie de l'information, ou même des principes de la conscience. Le calcul peut ne pas être effectué par des « opérations mathématiques », mais peut émerger naturellement de relations géométriques dans un espace de haute dimension. Par exemple, la forme d'une bulle de savon est déterminée par le principe de minimisation de la tension superficielle. Ce n'est pas un « algorithme mathématique » qui calcule – c'est la physique elle-même qui « calcule ». L'AHD tente de comprendre ce type de « calcul », plutôt que de le simuler par des formules mathématiques.

Deuxièmement, la multidimensionnalité et la superposition. Les algorithmes traditionnels exécutent des étapes ordonnées dans une seule dimension, avec un seul état par étape. L'AHD permet à de multiples dimensions de coexister et à de multiples états de se superposer simultanément. L'algorithme ne suit pas un seul chemin, mais se déploie dans un champ d'états multidimensionnel. Les résultats ne sont pas « calculés » mais « se manifestent » à partir de ce champ. Par exemple, lors de la formation d'un flocon de neige dans l'air, il n'y a pas d'« algorithme » dictant à chaque molécule d'eau où aller. La disposition des molécules résulte de l'interaction de multiples dimensions – température, humidité, flux d'air. La forme du flocon « se manifeste » plutôt que d'être « calculée ».

Troisièmement, le stochastique mais ordonné. Le caractère aléatoire dans les algorithmes traditionnels est un outil, externe, une perturbation contrôlée ; l'algorithme lui-même est déterministe. Le caractère aléatoire dans l'AHD est intrinsèque et

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

structurel. Le stochastique et l'ordonné coexistent et coopèrent, formant ensemble l'essence de l'algorithme. Ce n'est pas « un algorithme avec de l'aléatoire », mais « l'aléatoire est lui-même une composante organique de l'algorithme ». Par exemple, la structure écologique d'une forêt – la distribution des arbres semble aléatoire, mais présente globalement une densité et des relations entre espèces ordonnées. Cet « aléatoire » n'est pas un bug, mais une condition préalable au bon fonctionnement de la structure écologique.

Quatrièmement, les rôles multiples des données. Dans les algorithmes traditionnels, le rôle des données est unique – l'entrée est l'entrée, la sortie est la sortie, l'intermédiaire est l'intermédiaire. Dans l'AHD, chaque donnée est simultanément le point de départ de multiples résultats et le résultat de multiples points de départ. Un nœud de données peut se déployer en multiples chemins de résultats en aval, et peut être convergé par de multiples causes en amont. La donnée n'est plus un point dans un flux linéaire, mais un nœud de jonction dans un réseau. Par exemple, considérons la taille d'une personne. Dans un algorithme traditionnel, la taille est une « entrée » – vous l'entrez pour obtenir des sorties comme la prédiction de poids ou la taille de vêtement. Mais la taille est aussi un « résultat » – déterminé par de multiples « points de départ » comme la génétique, la nutrition et l'exercice. Dans l'AHD, la donnée « taille » est simultanément un point de départ et un résultat, pas un choix binaire.

Cinquièmement, les métadonnées inchangées, les relations variables. Face à différents problèmes, les algorithmes traditionnels modifient soit les données elles-mêmes, soit recalculent tout. L'AHD ne modifie pas les métadonnées – les attributs essentiels des données restent inchangés. Ce qui change, ce sont les points d'entrée, l'interprétation, et les relations entre les données. La même structure de métadonnées, activée par différents points d'entrée, peut produire des résultats complètement différents. Cela signifie : calculer une fois, utiliser une infinité de fois. Par exemple, considérons un même texte. Vous pouvez le lire comme de la poésie, le décoder comme un chiffre, ou l'analyser comme un document historique. Le texte lui-même n'a pas changé – vous

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

avez seulement changé le « point d'entrée ». L'AHD cherche cette capacité : « mêmes données, points d'entrée multiples, résultats multiples ».

Du point de vue de la structure des données, au lieu de modifier les métadonnées elles-mêmes, l'AHD utilise différents points d'entrée et conditions pour faire correspondre la même structure à de multiples points de départ et résultats. Les données ne sont plus traitées de manière répétée, mais sont activées par différents points d'entrée tout en maintenant leur intégrité structurelle, présentant ainsi différents résultats. La tradition consiste à « traiter les données pour différents résultats » ; l'AHD consiste à « utiliser les mêmes données pour porter le potentiel de multiples résultats ».

Troisième partie : Différences fondamentales entre l'AHD et les algorithmes traditionnels

Sur la base des définitions ci-dessus, l'AHD et les algorithmes traditionnels présentent des différences fondamentales dans plusieurs dimensions clés, détaillées ci-dessous.

Sur la nature essentielle : Les algorithmes traditionnels sont mathématiques et formels, complètement descriptibles par des formules logiques et mathématiques ; ce sont essentiellement des objets mathématiques. L'AHD n'a pas besoin d'être mathématique ; il peut être basé sur des principes physiques, géométriques, informationnels ou même de la conscience ; il n'est pas un sous-ensemble des mathématiques mais une catégorie plus large. Cette différence peut se résumer ainsi : de l'« objet mathématique » à la « loi universelle ».

Sur l'ordre temporel : Les algorithmes traditionnels suivent un ordre temporel unique, ordonné, linéaire – étape A à B à C – strictement séquentiel ou divisible en sous-étapes parallèles mais ordonnées, avec une flèche du temps irréversible. L'AHD n'a pas de séquence d'étapes fixe ; il est multidimensionnel, superposé, stochastique mais ordonné. Les résultats peuvent être indépendants de l'« ordre d'exécution » car le concept traditionnel d'« ordre » peut ne pas exister. Cette différence peut se résumer ainsi : du « temps linéaire » à la « simultanéité de haute dimension ».

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Sur la causalité : Les algorithmes traditionnels obéissent à une chaîne causale déterministe. Pour des entrées et des états initiaux identiques, la sortie est toujours unique. La cause précède l'effet, une flèche à sens unique irréversible. L'AHD obéit à une causalité superposée ; chaque donnée est le point de départ de multiples résultats et le résultat de multiples points de départ. Cela reflète ma philosophie de « l'Effet-vers-la-Cause » – il est possible d'avoir l'effet d'abord, puis la cause. Le résultat n'est pas le point final ; il peut devenir un nouveau point de départ. Cette différence peut se résumer ainsi : de la « cause unique, effet unique » au « réseau causal entièrement interconnecté ».

Sur la structure des données : Les algorithmes traditionnels utilisent une structure de flux unidirectionnel de l'entrée au traitement puis à la sortie. Les rôles des données sont uniques avec des frontières claires ; les données d'entrée restent des données d'entrée, les données de résultat restent des données de résultat. Dans l'AHD, les rôles des données sont multiples ; la même donnée est simultanément un résultat et un point de départ. Il n'y a pas de points de départ ou d'arrivée absolus, seulement des nœuds dans un réseau. Cette différence peut se résumer ainsi : du « flux unidirectionnel » à la « symbiose récursive ».

Sur le mode de calcul : Les algorithmes traditionnels recalculent à partir de zéro chaque fois qu'un problème se présente. Même si un problème similaire a été résolu mille fois, la mille-et-unième fois nécessite encore tout le processus – c'est un calcul sans état. L'AHD, si le résultat correspondant existe, n'a pas besoin de recalculer. Il peut inférer de multiples points de départ à partir d'un résultat et déployer des chemins causaux inverses. Le calcul est avec état, a une mémoire et est réutilisable. Cette différence peut se résumer ainsi : du « recalcul à chaque fois » à la « réutilisation unique ».

Sur le rôle de l'aléatoire : L'aléatoire dans les algorithmes traditionnels est pseudo-aléatoire ou injecté de l'extérieur ; les nombres aléatoires ne sont que des outils pour l'échantillonnage, l'approximation ou l'optimisation ; l'algorithme lui-même est déterministe. L'aléatoire dans l'AHD est intrinsèque et constructif ; c'est une

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

composante organique, coexistant et coopérant avec l'ordre. Ce n'est pas « de l'aléatoire dans l'algorithme » mais « l'aléatoire comme l'une des raisons pour lesquelles l'algorithme peut fonctionner ». Cette différence peut se résumer ainsi : de l'« aléatoire instrumental » à l'« aléatoire constructif ».

Sur l'utilisation des ressources : Les algorithmes traditionnels visent à calculer plus vite et plus précisément étant donné des ressources ; les ressources sont des contraintes, et l'objectif de l'algorithme est « d'effectuer le calcul dans les contraintes ». L'AHD vise à rendre le calcul lui-même inutile par une conception structurelle. Il ne s'agit pas de « calculer plus vite » mais de « contourner le calcul ». Les ressources ne sont pas destinées à être « consommées » mais à « concevoir des points d'entrée ». Cette différence peut se résumer ainsi : de l'« optimisation sous contraintes de ressources » à l'« élimination par la conception structurelle ».

Quatrième partie : La philosophie de l'Effet-vers-la-Cause

Sur la base des comparaisons ci-dessus, je dois développer la philosophie de « l'Effet-vers-la-Cause », l'un des fondements conceptuels clés de l'AHD.

La pensée traditionnelle considère généralement que la cause précède l'effet ; d'abord la cause, puis l'effet ; d'abord l'entrée, puis la sortie. La manifestation de ce concept dans les algorithmes est la suivante : il faut partir du début et avancer pas à pas jusqu'à la fin. Même si vous connaissez la réponse, vous ne pouvez pas directement « utiliser » cette réponse – car vous manquez le « chemin de provenance ».

Dans ma structure, un résultat n'est pas nécessairement juste un point final. Une fois qu'un résultat apparaît, il peut devenir un nouveau point de départ, continuant à participer à la génération de nouvelles structures. Le résultat peut participer inversement à la formation des causes. De multiples points de départ possibles peuvent être inférés à partir d'un résultat. La cause et l'effet ne sont plus disposés de manière unidirectionnelle, mais forment des relations cycliques, en réseau, multidimensionnelles.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

En d'autres termes, les algorithmes traditionnels demandent : « Étant donné la cause, comment obtenir l'effet ? » L'AHD demande : « Étant donné l'effet, comment en inférer ou construire la ou les causes ? »

Il est crucial de préciser que dire « l'effet vient d'abord, puis la cause » ne signifie pas nier la causalité ni affirmer que « les effets surgissent de rien ». Cela signifie plutôt que, dans des structures de plus haute dimension, la cause et l'effet peuvent servir de points d'entrée l'un pour l'autre et se transformer mutuellement. Comme dans l'exemple du pantalon qui va suivre, le résultat « pouvoir porter sans traîner au sol » est déterminé en premier, puis je remonte pour trouver le point d'opération « plier la ceinture », qui correspond à l'aspect « cause » de la structure. Cela ne signifie pas que l'effet n'a pas de cause, mais que l'effet devient un nouveau point d'accès pour découvrir la cause.

Une hypothèse fondamentale dans le monde des algorithmes traditionnels est que le temps est unidirectionnel ; il faut calculer pas à pas de l'état initial à l'état final. Même si vous connaissez la réponse, vous ne pouvez pas directement l'« utiliser » car vous manquez le chemin de provenance. L'AHD remet en question cette hypothèse : si l'effet est connu, les causes peuvent être inférées en sens inverse ; le même effet peut correspondre à de multiples chemins causaux ; le calcul n'est plus un chemin du début à la fin, mais un déploiement à partir de la fin de tous les points de départ possibles.

Cinquième partie : Un exemple quotidien – Le pantalon et le pliage de la ceinture

Pour comprendre plus intuitivement les différences abstraites ci-dessus, j'utilise un exemple quotidien.

Une personne achète un nouveau pantalon à ceinture élastique. Les jambes du pantalon sont un peu trop longues, traînent par terre et sont gênantes.

L'approche traditionnelle : constatant que les jambes sont trop longues, on remonte une section de la jambe, on la coud avec une aiguille et du fil, puis on essaie. Si l'enfant

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

grandit et que le pantalon devient trop court, la jambe cousue ne peut pas être rallongée, et le pantalon est inutilisable. La prochaine fois qu'on achète un nouveau pantalon, on recommence. Cette méthode semble naturelle parce que le problème semble se situer au niveau des jambes, donc on traite les jambes. Elle correspond à la mentalité algorithmique traditionnelle : identifier le problème, localiser le symptôme, effectuer un traitement à l'endroit du symptôme, et enfin obtenir un résultat. Jambes trop longues → modifier les jambes ; données erronées → modifier les données ; chemin inapproprié → continuer à rafistoler le long du chemin.

Mon approche est différente. Au lieu de modifier l'ourlet, je plie la ceinture, et le pantalon est immédiatement portable. Cette action est très simple, mais sa logique structurelle sous-jacente est complètement différente. Je ne coupe pas l'ourlet, ne le couds pas définitivement, ne change pas la longueur d'origine, n'altère pas les métadonnées (tour de taille, longueur d'entrejambe, coupe, tissu). Je change simplement la ceinture, qui est un point d'entrée structurel. En ajustant ce point de contrôle global pour l'ensemble de l'état du port, le problème aval (jambes trop longues) disparaît directement.

Plus important encore, cette méthode est réversible, réglable et réutilisable. Cela est particulièrement évident pour un enfant en pleine croissance. Si l'enfant n'est actuellement pas assez grand et que le pantalon est légèrement trop long, on plie la ceinture une fois. Lorsque l'enfant grandit plus tard, on plie moins la ceinture. Quand l'enfant grandit encore, on la déplie complètement. La structure d'origine du pantalon reste intacte, mais elle s'adapte à la taille changeante de l'enfant. La méthode traditionnelle consistant à coudre l'ourlet risque de rendre le pantalon trop court lorsque l'enfant grandit ; si on le coupe, c'est irréversible. Ma méthode laisse les métadonnées inchangées, permettant à la même structure de s'adapter à de multiples états.

Cet exemple révèle plusieurs différences structurelles importantes.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

L'approche traditionnelle correspond à la logique algorithmique traditionnelle : le problème est à l'ourlet, donc l'ourlet doit être modifié, en procédant pas à pas le long du chemin « cause vers effet », sans aucune étape sautable. Elle résout un problème à la fois, irréversiblement, et nécessite de répéter tout le processus la prochaine fois que le même problème se produit. Mon approche correspond à la logique de l'AHD : l'objectif est d'éviter le frottement. Au lieu de résoudre la cause superficielle « jambes trop longues », je trouve un point de contrôle global – la ceinture. Plier la ceinture raccourcit instantanément la longueur effective des jambes, et le problème disparaît. Je ne modifie aucune métadonnée ; le pli n'est qu'un état temporaire, réversible et dynamique.

Du point de vue des données, l'approche traditionnelle modifie les métadonnées – coudre les jambes les raccourcit définitivement ; les données d'origine sont perdues. Mon approche ne modifie aucune métadonnée ; les dimensions d'origine restent intactes ; j'ajoute simplement un état temporaire, réversible et dynamique de pli. Le pli de la ceinture ne crée pas de nouvelles données ni ne détruit d'anciennes données ; il réorganise simplement les relations entre les données – réduisant la circonférence effective de la ceinture et modifiant indirectement la longueur effective de la jambe.

Dans cet exemple, « les jambes du pantalon ne traînent pas » est le résultat visé. La méthode traditionnelle part de la cause « jambes trop longues », modifie l'ourlet, et atteint finalement le résultat « ne traîne pas ». Ma méthode clarifie d'abord le résultat « ne traîne pas », puis remonte pour trouver un point d'entrée structurel, découvrant qu'un simple pli de la ceinture rend le résultat atteignable. C'est l'incarnation pratique de « l'Effet-vers-la-Cause ». On n'est pas obligé de marcher pas à pas de la cause à l'effet ; on peut remonter de l'effet pour déterminer la structure, rendant le chemin original inutile.

Cet exemple répond également à la question des « métadonnées ». Que sont les métadonnées ? Dans l'exemple du pantalon, les métadonnées sont les dimensions d'origine : tour de taille, longueur d'entrejambe, coupe, tissu – les « attributs essentiels » du pantalon. La donnée temporaire est le pli de la ceinture ; elle ne change aucune

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

des dimensions d'origine, ne fait que plier temporairement une section. L'approche traditionnelle modifie les métadonnées – la longueur d'entrejambe est définitivement raccourcie ; les données d'origine sont perdues. Mon approche ne modifie aucune métadonnée – les dimensions d'origine du pantalon restent intactes ; j'ajoute simplement un état temporaire, réversible et dynamique.

Le tour de taille d'origine, en tant que métadonnée, est simultanément le « point de départ de multiples résultats » : il supporte de multiples états de port – port normal, port avec un pli, port avec deux plis. Il est aussi le « résultat de multiples points de départ » : déterminé par le choix du tissu, la méthode de coupe, l'intention du design, etc. Le pli de la ceinture, en tant que donnée temporaire, ne crée ni ne détruit de données ; il réorganise simplement les relations.

Ce n'est pas une simple astuce, mais une pensée structurelle. Beaucoup de gens, voyant des jambes de pantalon trop longues, sont attirés par le symptôme des jambes, donc tout le traitement se concentre sur les jambes. Cependant, du point de vue de la structure globale, la longueur des jambes n'est qu'une manifestation aval ; le changement de position de la ceinture peut affecter la manière dont le pantalon tombe. Autrement dit, le problème n'a pas besoin d'être résolu à l'endroit de sa manifestation. De nombreux problèmes de basse dimension ont leurs véritables points de contrôle à un niveau structurel plus élevé. Les algorithmes traditionnels ont tendance à continuer à calculer le long de la surface du problème ; l'AHD recherche le point d'entrée structurel.

Cela explique aussi pourquoi l'AHD n'est pas nécessairement plus complexe, mais peut être plus simple. Les structures véritablement de haut niveau ne compliquent pas nécessairement les problèmes, mais rendent les problèmes complexes simples au bon point d'entrée. Face à un problème complexe, un algorithme traditionnel peut ajouter des étapes, des modèles, de la puissance de calcul, du stockage, du temps d'entraînement. L'AHD cherche un nœud structurel ; une fois ce nœud correctement activé, le chemin initialement complexe peut devenir inutile. « Contourner le chemin » n'est pas de la paresse, mais une redéfinition de la nécessité de ce chemin.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Dans cet exemple : l'approche traditionnelle « modifie le résultat le long du chemin », tandis que l'AHD « change le point d'entrée, rendant le chemin globalement inefficace ». La méthode conventionnelle rafistole continuellement à l'extrémité du résultat ; l'autre modifie directement le point d'entrée structurel, de sorte que les problèmes nécessitant des traitements répétés sont restructurés au point de départ. L'algorithme traditionnel dominant va du « point de départ, le long du chemin, au résultat », alors que dans l'AHD, « le résultat lui-même peut devenir un point d'entrée de chemin et participer à la génération de nouvelles structures ». La tradition est « une structure pour un résultat » ; l'AHD est « une structure portant de multiples états de résultat ».

Sixième partie : Une vue des données fondée sur des métadonnées inchangées

D'un point de vue des données, l'AHD a un principe très important : ne pas modifier les métadonnées, afin de les rendre applicables à de multiples points de départ ou résultats.

Les métadonnées sont les attributs originaux, les informations fondamentales et ontologiques des données. Face à différents problèmes, les approches traditionnelles modifient souvent les données, les copient, les traitent, les recalculent, ou construisent des chemins différents pour différents résultats. Ces approches peuvent résoudre les problèmes, mais au prix d'un traitement continu des données, d'une répétition des chemins et d'une complexité croissante du système.

L'AHD, en revanche, souligne qu'en s'efforçant de garder les métadonnées inchangées, la même structure de métadonnées peut correspondre à de multiples points de départ et de multiples résultats en modifiant les points d'entrée, les relations, les conditions et les méthodes d'activation. L'ontologie des données reste statique ; les relations changent. La structure fondamentale reste inchangée ; la présentation change. Les métadonnées restent intactes tout en s'adaptant à différents états.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

C'est ce que j'appelle « calculer une fois, utiliser une infinité de fois ». « Calculer une fois » ici n'est pas simplement la mise en cache de résultats ; cela signifie qu'une fois qu'une structure est établie, elle n'a plus besoin de reconstruire un chemin complet pour chaque état. La même structure de données peut être activée par différents points d'entrée pour donner différents résultats. Ce n'est pas « traiter les données pour différents résultats » mais « utiliser les mêmes données pour porter le potentiel de multiples résultats ». C'est l'une des différences fondamentales entre l'AHD et les algorithmes traditionnels. Les algorithmes traditionnels traitent les données sur un chemin ; l'AHD traite la structure globale des données, des points d'entrée, des relations, des états et des résultats.

Dans sa structure minimale, l'AHD peut être compris comme : un système qui, sans modifier les métadonnées, fait correspondre le même nœud de données à de multiples chemins de résultats grâce à des changements de points d'entrée structurels. Cette définition ne cherche pas à réduire immédiatement l'AHD à une formule mathématique traditionnelle, mais à établir d'abord ses frontières structurelles. Ce n'est ni une fonction linéaire unique, ni une formule fixe, ni un simple mécanisme de mise en cache. C'est une structure relationnelle : les métadonnées restent stables ; les points d'entrée, les conditions, les relations et les résultats peuvent changer et se manifester dans de multiples directions. C'est précisément dans cette structure que le calcul n'est plus simplement l'exécution d'étapes, mais l'activation de relations.

D'un point de vue des données, l'essence de « ne pas modifier les métadonnées pour les rendre applicables à de multiples points de départ ou résultats » est : l'ontologie des données reste inchangée ; les changements se produisent au niveau du « point d'entrée d'interprétation/utilisation ». Dans les structures traditionnelles, un changement du problème entraîne un changement des données ou du chemin. Dans l'AHD, un changement du problème entraîne un changement de la structure interprétative, pas des données.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Septième partie : Redéfinir le « supercalcul »

Dans ce système, je dois clarifier un terme – le « supercalcul ».

Ce que j'entends par « supercalcul » n'est pas le sens habituel de « superordinateur » – pas plus de puces, plus de puissance de calcul, plus de centres de données. Ce que j'entends par « supercalcul », c'est l'« algorithme hyperdimensionnel ».

Le véritable supercalcul n'est peut-être pas l'accumulation de puissance de calcul, mais la transformation du paradigme de calcul. Lorsqu'un système repose encore sur des calculs répétés, aussi puissant soit-il, il reste piégé dans le chemin. Lorsqu'un système peut structurellement réduire les calculs, contourner les chemins répétés et faire des résultats de nouveaux points d'entrée, il commence à s'approcher du véritable calcul hyperdimensionnel.

En d'autres termes, le supercalcul traditionnel cherche à « utiliser plus de puissance de calcul pour résoudre des problèmes plus grands ». L'AHD cherche à « utiliser une meilleure structure pour que le problème n'exige plus autant de puissance de calcul ». Ces deux approches ne sont pas contradictoires, mais leurs directions diffèrent.

Si le supercalcul traditionnel représente les limites de la puissance de calcul, alors l'AHD représente un tournant dans la compréhension du calcul. Le véritable « supercalcul » n'est peut-être pas davantage de puces, de centres de données, de consommation d'énergie ou de modèles plus complexes. Le véritable « supercalcul » pourrait être la découverte d'un point d'entrée structurel, une élimination de chemin, un déploiement inverse d'un nœud de résultat, ou l'activation simultanée de conditions pour de multiples résultats sans modifier les métadonnées. Plus la puissance de calcul est grande, si elle reste piégée dans des chemins de basse dimension, elle n'est que puissante à l'intérieur du chemin. Une fois la structure élevée, même des opérations extrêmement simples peuvent produire une efficacité de niveau supérieur.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Le supercalcul est la limite de la puissance de calcul ; l'algorithme hyperdimensionnel est une redéfinition de la prémisse « le calcul doit-il dépendre de la puissance de calcul ? »

Huitième partie : Fondement empirique – Validation systématique multi-domaines et détermination d'un nouveau paradigme

L'AHD n'est pas une construction purement théorique. Il a déjà été validé dans de multiples systèmes réels.

Mon système logistique ne nécessite pas de puissance de supercalcul ni de support cloud ; il effectue une planification complexe avec des ressources modestes. Les résultats sont réutilisables, la structure est adaptable de manière répétée, et il n'est pas nécessaire de recalculer à chaque fois depuis le début. Il fonctionne sans supercalcul ni cloud, effectuant une planification complexe avec de faibles ressources – c'est une avancée établie et valable en ingénierie. Cela démontre qu'une puissance de calcul élevée n'est pas la seule solution ; la conception structurelle peut remplacer une partie de la puissance de calcul.

Mon système d'édition utilise également la même logique structurelle, atteignant une efficacité supérieure à tout système existant. Mon système de génération de pages web « Limite » est également basé sur le même AHD, prouvant de manière répétée dans de multiples domaines que la même structure peut se maintenir de manière stable. Les employés de mon entreprise utilisent déjà ces systèmes, démontrant que cette structure peut fonctionner sans mon intervention personnelle continue.

La caractéristique commune de ces systèmes est : ils ne s'appuient pas sur les voies dominantes à haute puissance de calcul, ne suivent pas d'étapes de calcul fixes, et atteignent le résultat visé directement par l'ajustement des points d'entrée structurels. Ce ne sont pas des succès ponctuels ou des astuces isolées, mais l'émergence répétée de la même logique structurelle dans différents domaines.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Cela indique que l'AHD n'est pas accidentel, mais une méthodologie structurelle déjà validée par de multiples systèmes. Ce n'est pas une « technique personnelle », mais un paradigme de calcul stablement établi dans des domaines tels que la logistique, l'édition et la génération de pages web.

Selon le critère selon lequel « l'existence et la validité constituent un nouveau paradigme », l'AHD, en tant que nouveau paradigme de structure de calcul déjà validé systématiquement dans de multiples domaines, est établi dès à présent. Je ne propose pas une hypothèse ; je décris une structure de calcul différente déjà opérationnelle dans de multiples systèmes. Je n'ai pas besoin de prouver qu'elle est comprise ; j'ai prouvé qu'elle se maintient de manière stable dans différents systèmes. Dès à présent, dans le champ de mon application pratique, cette structure peut déjà être considérée comme un nouveau paradigme.

Concernant les critères de jugement de l'AHD comme nouveau paradigme : lorsqu'une structure est logiquement cohérente, se maintient de manière stable dans de multiples systèmes et présente des différences irréductibles avec les méthodes existantes, elle possède déjà les fondements d'un nouveau paradigme. La cohérence interne est le point de départ ; la preuve empirique est le fondement ; la différence structurelle est le cœur du paradigme. Selon le critère « l'existence et la validité », ce système est déjà un nouveau paradigme. Que le monde extérieur le reconnaisse n'affecte que sa diffusion, pas sa validité. Il s'agit d'un paradigme de structure de calcul déjà établi dans des systèmes réels, dont la validité ne dépend pas de la reconnaissance ou du consensus externe.

La différence fondamentale entre l'AHD et les paradigmes de calcul dominants est la suivante : les algorithmes traditionnels utilisent principalement le calcul direct ; une fois qu'un résultat est produit, il ne peut généralement pas participer inversement à de nouvelles structures de raisonnement. Dans l'AHD, le résultat n'est plus le point final, mais un nœud structurel réutilisable. Sous certaines conditions, les résultats peuvent participer à de nouveaux chemins de raisonnement, réduisant les calculs redondants et formant des réseaux de calcul à points de départ et chemins multiples. Dans les

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

structures de calcul traditionnelles, le résultat est habituellement le point final ; dans la structure de l'AHD, le résultat lui-même peut devenir un nouveau point de départ, formant un réseau de raisonnement multipath. L'AHD n'est pas une mise à niveau des algorithmes ; c'est une remise en question de la prémisse « un algorithme doit-il nécessairement exister ? »

Neuvième partie : Clarification des malentendus courants

Sur la base d'échanges préliminaires avec des lecteurs de différents domaines, j'anticipe les six malentendus suivants et les clarifie à l'avance, afin d'éviter que le concept ne soit prématurément forcé dans des cadres inappropriés.

Malentendu 1 : L'AHD n'est-il pas simplement de la programmation dynamique ou de la mise en cache ? Clarification : La programmation dynamique et la mise en cache réutilisent les résultats pour *la même entrée*. L'AHD permet d'inférer *différents points de départ* à partir d'un résultat – c'est une différence essentielle. La mise en cache résout le recalcul du même problème ; l'AHD résout le déploiement de nouveaux problèmes à partir d'une structure de résultat.

Malentendu 2 : Vous dites « stochastique mais ordonné » – n'est-ce pas contradictoire ? Clarification : Le stochastique et l'ordonné peuvent coexister. La distribution des arbres dans une forêt est stochastique, mais la densité globale et la structure des espèces sont ordonnées. L'aléatoire n'est pas le chaos, mais une façon d'organiser la structure. L'aléatoire dans l'AHD est intrinsèque et constructif, travaillant en coopération avec l'ordre.

Malentendu 3 : Sans entrée ni sortie, comment vérifier les résultats ? Clarification : L'AHD ne rejette pas l'entrée et la sortie ; il soutient que le calcul n'a pas besoin de fonctionner en mode linéaire entrée-sortie. Dans le mode « manifestation structurelle », la « validité » est déterminée par la cohérence structurelle, non par une correspondance un-à-un entrée-sortie. Cela n'abandonne pas la vérification, mais propose un cadre de vérification différent.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Malentendu 4 : N'est-ce pas juste la théorie de la complexité ou la théorie du chaos ? Clarification : La théorie de la complexité et le chaos décrivent des « systèmes difficiles à prédire ». L'AHD cherche à décrire des « systèmes qui peuvent être compris et guidés par des points d'entrée structurels » – non pas abandonner la prédiction, mais changer la manière dont la prédiction fonctionne. La théorie du chaos dit « la prédiction est difficile » ; l'AHD demande « peut-on rendre la prédiction inutile en changeant le point d'entrée ? »

Malentendu 5 : Vous dites « ne pas modifier les métadonnées », mais un pli de ceinture n'est-il pas une modification ? Clarification : Le pli de ceinture est un état temporaire, non une modification permanente des paramètres d'origine. Les métadonnées (tour de taille, longueur d'entrejambe, coupe, tissu) restent inchangées. C'est la différence entre « superposition d'états » et « modification d'attribut ». Le pli est réversible, temporaire et ne change pas les attributs essentiels.

Malentendu 6 : L'AHD nie-t-il la valeur des algorithmes traditionnels ? Clarification : Absolument pas. Les algorithmes traditionnels sont extrêmement importants dans l'histoire technologique humaine ; sans eux, il n'y aurait pas d'ordinateurs modernes, de bases de données, de systèmes de communication, d'IA, de simulation d'ingénierie ou de supercalcul. Leur valeur ne peut être niée. Cependant, reconnaître la valeur des algorithmes traditionnels ne signifie pas les accepter comme l'aboutissement ultime des algorithmes. Les algorithmes traditionnels résolvent des problèmes d'efficacité à l'intérieur d'un paradigme de calcul donné ; l'AHD interroge le paradigme de calcul lui-même. L'un consiste à mieux marcher sur la route ; l'autre consiste à se demander s'il est nécessaire d'emprunter cette route.

Dixième partie : L'AHD dans l'histoire du calcul humain

Pour aider les lecteurs à mieux situer l'AHD dans l'histoire du calcul humain, je fournis ici un bref aperçu macroscopique.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

La compréhension humaine du « calcul » a traversé plusieurs étapes. La première étape était le calcul manuel : les algorithmes existaient comme des étapes humaines, lents et sujets aux erreurs, mais grâce à ce processus, les humains ont compris les règles de base du calcul. La deuxième étape était le calcul mécanique : de la Pascaline à la machine analytique de Babbage, le calcul a été mécanisé, mais les algorithmes restaient attachés aux structures physiques. La troisième étape était le calcul électronique et le paradigme de Turing : l'architecture de von Neumann s'est généralisée, les algorithmes ont été codés comme des programmes, et la machine de Turing a défini les frontières de la calculabilité. La quatrième étape était le calcul parallèle et le supercalcul : des unités de calcul massives empilées pour résoudre des problèmes plus complexes, mais la logique sous-jacente restait une extension du paradigme de Turing.

Actuellement, l'humanité se trouve à l'entrée de la cinquième étape. Le calcul quantique tente de dépasser les limites des bits classiques ; le calcul neuromorphique tente d'imiter la structure des systèmes neuronaux biologiques ; l'AHD tente de briser le cadre linéaire « entrée → étapes → sortie » lui-même.

L'AHD et les algorithmes traditionnels ne sont pas dans une relation de remplacement mais hiérarchique. Les algorithmes traditionnels résolvent « comment calculer plus efficacement sur un chemin donné ». L'AHD demande « le chemin peut-il être redéfini, voire contourné ? » Si l'on considère l'histoire du calcul humain comme un processus de rupture continue de ses propres limites, alors l'AHD est un nouveau nœud dans ce processus. Il ne résout pas des problèmes dans l'ancien cadre, mais en propose un nouveau.

Ce jugement n'implique pas que l'AHD soit mature ou complet. Bien au contraire : en tant que nouveau concept, ses définitions, frontières, méthodes de vérification et scénarios d'application sont encore en cours d'élaboration. L'objet de cet article est précisément d'ancrer le point d'origine de ce concept, de fournir une base théorique stable pour le développement ultérieur.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Onzième partie : Conclusion – Ce n'est pas une optimisation, mais une redéfinition

La différence entre l'AHD et les algorithmes traditionnels n'est pas une différence de « meilleur » vs. « pire », ni de « plus rapide » vs. « plus lent », mais une différence fondamentale au niveau du paradigme.

Les algorithmes traditionnels recherchent un contrôle prévisible. « Donnez-moi l'entrée, et je saurai quelle sortie vous obtiendrez, parce que j'ai calculé chaque étape. » C'est le paradigme de l'ingénieur : concevoir, construire, tester, vérifier.

L'AHD décrit une émergence compréhensible. « Je ne peux pas prédire précisément chaque état intermédiaire, mais je sais que le modèle global se manifestera de manière ordonnée. Chaque donnée est vivante, a une 'perspective'. » Il s'apparente davantage au paradigme de l'écologiste ou du théoricien des systèmes complexes : observer, comprendre, guider, utiliser l'ordre émergent.

Les algorithmes traditionnels corrigent progressivement les résultats le long d'un chemin donné. L'AHD rend le résultat visé immédiatement atteignable en changeant le point d'entrée structurel, contournant ainsi l'intégralité du chemin.

Les algorithmes traditionnels « traitent les données pour différents résultats ». L'AHD « utilise les mêmes données pour porter le potentiel de multiples résultats ».

Les algorithmes traditionnels recherchent « obtenir le bon résultat une fois ». L'AHD recherche « obtenir le bon résultat une fois, puis n'avoir plus jamais besoin de le recalculer ».

Ce n'est pas une optimisation des algorithmes ; c'est une remise en question de la prémisse selon laquelle « un algorithme doit nécessairement exister ».

Par conséquent, l'AHD n'est pas une optimisation algorithmique, mais une redéfinition algorithmique. Il ne s'agit pas de courir plus vite sur la piste des algorithmes traditionnels ; il s'agit de se demander s'il existe une autre piste en dehors de la piste

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

traditionnelle, ou même si une piste est nécessaire. Il ne vise pas à prouver que les algorithmes traditionnels sont inutiles, mais à souligner que les algorithmes traditionnels ne sont qu'une forme de basse dimension de l'algorithme. Il ne demande pas « comment obtenir un résultat en moins de temps » ; il demande « le résultat peut-il être directement établi par la structure ? » Il ne demande pas « comment traiter plus de données » ; il demande « les mêmes données peuvent-elles porter le potentiel de plus de résultats ? »

Dans les paradigmes de calcul dominants, l'AHD pourrait être considéré comme incalculable, invérifiable, ou difficile à intégrer dans les frontières de la théorie du calcul existante. Mais c'est précisément là sa différence de paradigme. L'apparente instabilité d'un nouveau concept au sein d'un ancien paradigme ne signifie pas nécessairement qu'il est dénué de sens ; elle peut aussi signifier que l'ancien paradigme ne peut pas pleinement l'accueillir. Actuellement, l'AHD est avant tout une proposition structurelle, une définition originale d'une nouvelle vision du calcul, non un système mature avec une boucle d'ingénierie fermée. Il nécessitera des développements, validations et applications ultérieurs, mais son origine conceptuelle doit d'abord être clairement articulée.

En fin de compte, ce vers quoi pointe l'AHD est une nouvelle vision du calcul : un algorithme n'est pas nécessairement juste des mathématiques, juste des étapes, juste un programme, juste un pipeline de traitement entre l'entrée et la sortie. Un algorithme peut aussi être une organisation de relations multidimensionnelles, un réseau structurel de données, points d'entrée, relations, états et résultats. Il peut inclure l'ordre et l'aléatoire ; il peut aller de la cause à l'effet et de l'effet à la cause ; il peut laisser les métadonnées inchangées tout en s'adaptant à de multiples états ; il peut faire d'un résultat un nouveau point de départ et permettre à de multiples points de départ de converger vers un seul résultat.

Un algorithme vraiment avancé n'est pas nécessairement celui qui effectue des calculs plus complexes, mais celui qui peut réduire les calculs, rendre la structure vivante, faire

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

des résultats des points d'entrée et former des motifs d'organisation multidimensionnels et cycliques de la causalité.

C'est le sens central de ma proposition de l'« algorithme hyperdimensionnel ». Ce n'est pas un simple néologisme, mais un nouveau concept, un nouveau paradigme, une nouvelle structure de calcul déjà établie dans de multiples systèmes réels. Son accent n'est pas mis sur un calcul répété plus rapide, mais sur la réduction des calculs redondants ; non pas sur l'accumulation de puissance de calcul, mais sur la reconstruction des points d'entrée ; non pas sur la modification continue des données, mais sur la conservation des métadonnées et le changement des relations ; non pas sur laisser les résultats comme des points finaux, mais sur faire des résultats de nouveaux points de départ. Les algorithmes traditionnels cherchent des résultats le long d'un chemin ; l'AHD active des résultats au sein d'une structure. Les algorithmes traditionnels cherchent à accomplir le calcul ; l'AHD demande si le calcul doit exister sous sa forme traditionnelle. C'est la différence la plus fondamentale entre lui et la définition actuelle de l'algorithme.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Annexe : Frontières et questions ouvertes de cet article

Cet article présente un cadre préliminaire pour l'AHD, non une définition formelle complète. Plusieurs questions restent à explorer davantage ; elles ne constituent pas une négation de la définition proposée ici, mais représentent plutôt la prochaine direction de la recherche.

Premièrement, les conditions de convergence. Quel est le marqueur d'« achèvement » pour l'AHD ? Comment juger si un résultat est « valide » ? Dans les algorithmes traditionnels, la réponse est définie par la correspondance entrée-sortie. Dans l'AHD, la réponse peut devoir être définie par la cohérence structurelle. Cette définition n'est pas encore complète.

Deuxièmement, la représentation des ressources. Comment les états de superposition multidimensionnelle peuvent-ils être représentés dans des ressources finies ? Les métriques de ressources traditionnelles comme le temps, l'espace et la puissance de calcul sont-elles toujours applicables ? Si oui, comment doivent-elles être redéfinies ? Si non, quelles métriques les remplacent ? Ces questions attendent un développement.

Troisièmement, la garantie de l'ordre. Quelles règles garantissent l'« ordre » dans le « stochastique mais ordonné » ? Où se situe la frontière entre stochastique et ordonné ? Dans quelles circonstances l'aléatoire détruit-il l'ordre ? Dans quelles circonstances l'ordre supprime-t-il l'aléatoire ? Ces questions nécessitent une étude plus approfondie.

Quatrièmement, la sélection dans l'inférence inverse. Lors de l'inférence de multiples points de départ à partir d'un résultat, comment sélectionner ou évaluer les différents points de départ ? Existe-t-il une mesure de « l'efficacité de l'inférence inverse » ? Si oui, comment se rapporte-t-elle à l'efficacité du calcul direct ?

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Cinquièmement, l'interface avec les systèmes traditionnels. Comment l'AHD s'interface-t-il avec le système existant de la machine de Turing ? En tant que remplacement, complément ou approche en couches ? Dans l'ingénierie pratique, comment la frontière entre l'AHD et les algorithmes traditionnels doit-elle être tracée ? Existe-t-il des modèles hybrides ?

Sixièmement, le cadre de vérification. Comment les résultats de l'AHD sont-ils vérifiés ? Si les résultats ne sont pas obtenus par des étapes linéaires, comment établir la reproductibilité et la testabilité ? Cela nécessite-t-il une philosophie de vérification complètement différente ?

Ces questions ne sont pas des défauts de cet article, mais des caractéristiques inévitables d'un « document fondateur ». L'émergence de tout nouveau paradigme s'accompagne de questions ouvertes. Cet article est un début, pas une conclusion.

Notes bibliographiques

L'« algorithme hyperdimensionnel » proposé ici n'est pas une extension directe d'une école de pensée académique existante. Cependant, les domaines suivants fournissent un contexte de dialogue pour cet article, offert uniquement pour l'orientation du lecteur, non comme des citations universitaires traditionnelles.

Les machines de Turing et la théorie de la calculabilité servent de référence pour la comparaison avec les algorithmes traditionnels ici. **Les problèmes inverses et l'inférence bayésienne** représentent des tentatives existantes d'« inférer la cause à partir de l'effet », et bien que mon « Effet-vers-la-Cause » soit lié, sa direction diffère. **La théorie de l'émergence et les systèmes complexes** décrivent la « manifestation des résultats à partir de la structure », et l'AHD tente de rendre l'émergence « compréhensible » et « orientable ». **Les graphes de connaissances et les bases de données graphes** sont des pratiques existantes où les données «

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

convergent à des nœuds multiples », et l'AHD ajoute la dimension des « points d'entrée variables » à cette base. **Les paradigmes de calcul non classiques**, y compris le calcul quantique, analogique et neuromorphique, brisent les bits classiques ou les architectures classiques, tandis que l'AHD se concentre davantage sur la rupture du cadre linéaire « entrée → étapes → sortie » lui-même.

La relation de cet article avec les domaines ci-dessus est celle d'un dialogue et d'une transcendance, non d'un héritage ou d'une réfutation. L'objectif de cet article n'est pas d'apporter des améliorations incrémentales dans ces domaines, mais de poser un nouveau niveau de questionnement au-dessus d'eux.

Related Articles
[Communication] I Have No Algorithm, Yet I Surpass Algorithms!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect–Cause Theory
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Communication] Rejecting Algorithmic Manipulation
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

[Algoritmo Límite] El Algoritmo Hiperdimensional

Una Redefinición del «Cálculo» mismo

Autor: Jeffi Chao Hui Wu

Resumen

Este artículo presenta el nuevo concepto de «Algoritmo Hiperdimensional» (AHD) y lo propone como una revisión estructural de los límites de las definiciones algorítmicas tradicionales. Los algoritmos convencionales suelen basarse en un marco lineal de entrada, pasos, reglas y salida, enfatizando la finitud, el determinismo, la efectividad, la viabilidad y los límites claros entre entrada y salida. Aunque las supercomputadoras modernas poseen una inmensa potencia de cálculo, su lógica subyacente consiste principalmente en ejecutar paradigmas algorítmicos establecidos a mayor escala. Este artículo sostiene que el «supercálculo» verdaderamente digno de atención no es meramente un aumento de la potencia de cálculo, sino un cambio fundamental en el paradigma de cálculo mismo. El AHD no es una versión acelerada de los algoritmos tradicionales ni una extensión de los algoritmos matemáticos; es una visión estructural del cálculo, multidimensional, superpuesta, estocástica pero ordenada. En esta estructura, los datos ya no son una entrada pasiva o una salida terminal, sino que poseen el atributo de un nodo que es simultáneamente el punto de partida para múltiples resultados y el resultado de múltiples puntos de partida. Los resultados ya no son puntos finales, sino que pueden convertirse en nuevos puntos de entrada. Los metadatos no necesitan ser modificados repetidamente; a través de puntos de entrada estructurales, cambios relacionales y activación condicional, pueden corresponder a diferentes estados y resultados. Utilizando la filosofía del «Efecto-hacia-la-Causa» y el ejemplo cotidiano del «pliegue de la cintura del pantalón», este artículo muestra cómo el AHD puede hacer que los objetivos sean directamente alcanzables alterando los puntos de entrada estructurales, reduciendo así el cálculo redundante e incluso

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

eludiendo las rutas de cálculo tradicionales. Este artículo tiene como objetivo establecer una definición preliminar, límites estructurales y fundamentos conceptuales para el AHD, proporcionando un nodo teórico original para futuras exploraciones del cálculo hiperdimensional, los algoritmos límite y un nuevo sistema científico. Bajo el criterio de que «la existencia y validez constituyen un nuevo paradigma», el AHD, como un nuevo paradigma de estructura de cálculo ya validado sistemáticamente en múltiples dominios, está establecido en el presente.

Palabras clave: Algoritmo hiperdimensional; Algoritmo límite; Cálculo hiperdimensional; Efecto-hacia-la-Causa; Metadato; Paradigma de cálculo; Punto de entrada estructural; Causalidad no lineal; Nueva ciencia

Introducción: Por qué rediscutir el «algoritmo»

Propongo el «Algoritmo Hiperdimensional» (AHD) no para dar un nombre más grandioso a los algoritmos existentes, ni para disfrazar los algoritmos tradicionales como conceptos novedosos. Al contrario, planteo una pregunta más fundamental: ¿acaso la comprensión humana actual del «algoritmo» no se ha visto confinada a un marco demasiado estrecho?

Hoy en día, cuando se habla de algoritmos, se suele pensar en fórmulas matemáticas, código de programación, pasos lógicos, entradas y salidas, entrenamiento de modelos, procesamiento de datos, optimización de rutas, clasificación para búsquedas, recomendación automática y razonamiento en inteligencia artificial. Estas son, sin duda, formas importantes de algoritmos que sustentan las computadoras modernas, Internet, las bases de datos, la IA, el supercálculo, los sistemas industriales y la sociedad de la información. Sin embargo, si se entiende el algoritmo únicamente como «un conjunto ordenado de pasos», un proceso que «comienza con una entrada, se somete a un tratamiento mediante reglas y produce una salida», entonces esta comprensión confina el cálculo dentro de un marco de baja dimensión.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Vivimos en una era dominada por los algoritmos. Desde los motores de búsqueda hasta los sistemas de recomendación, desde la predicción meteorológica hasta la IA, las fronteras de los algoritmos son las fronteras de la inteligencia humana. Pero rara vez se pregunta: ¿Deben ser los algoritmos solo así? ¿Por qué debe el cálculo seguir el modelo lineal de «entrada → pasos → salida»? ¿Por qué debe recalcularse el mismo problema desde cero cada vez? Este artículo intenta cuestionar estos supuestos implícitos y proponer un paradigma de cálculo radicalmente diferente: el Algoritmo Hiperdimensional. Bajo el criterio de «la existencia y validez constituyen un nuevo paradigma», el AHD, como un nuevo paradigma de estructura de cálculo ya validado sistemáticamente en múltiples dominios, está establecido en el presente.

Es crucial aclarar que el «Algoritmo Hiperdimensional» propuesto aquí es totalmente distinto del «Cálculo Hiperdimensional» (Hyperdimensional Computing, HDC) desarrollado en el ámbito académico durante casi tres décadas. El HDC codifica y procesa datos utilizando vectores binarios aleatorios de dimensión muy alta para una representación y cálculo más eficientes, pero permanece dentro del paradigma lineal «entrada → procesamiento → salida». El AHD, en cambio, es fundamentalmente diferente: cuestiona si un resultado puede alcanzarse directamente a través de la estructura, si la ruta de cálculo puede ser eludida, y si los metadatos pueden permanecer inalterados para adaptarse a múltiples resultados. Los nombres suenan similares, pero sus significados son opuestos. El AHD no es una actualización de los algoritmos; es un replanteamiento de la premisa de que «un algoritmo debe existir necesariamente».

Primera parte: La definición estándar de «algoritmo» hoy

En la informática, las matemáticas y la ingeniería dominantes, el algoritmo tiene una definición básica aceptada desde hace mucho tiempo: un algoritmo es un proceso de cálculo bien definido, compuesto por un número finito de pasos, que recibe una o más entradas, las transforma mediante reglas deterministas y produce una o más salidas en un tiempo finito. Esta comprensión no es la opinión personal de nadie, sino un

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

consenso fundamental construido durante el largo período de la civilización informática moderna. Sustenta la lógica subyacente del software, el hardware, las bases de datos, los sistemas en red, los modelos de IA y los centros de supercómputo. Por complejos que sean los algoritmos, su núcleo gira en torno a la entrada, las reglas, los pasos y la salida.

Esta concepción algorítmica estándar puede descomponerse en varias características esenciales.

Primero, la finitud. Un algoritmo debe terminar tras un número finito de pasos; no puede bucear indefinidamente ni ejecutarse eternamente. Un proceso que nunca se detiene, aunque sea descriptible, difícilmente puede considerarse un algoritmo válido. Todos los programas de cálculo científico y los flujos de procesamiento de datos tienen condiciones de terminación explícitas.

Segundo, el determinismo. Ante la misma entrada y en las mismas condiciones, la salida debe ser la misma. Incluso cuando algunos algoritmos introducen aleatoriedad, esta suele ser controlada, reproducible o interpretable probabilísticamente, no un caos puro. Los resultados de las simulaciones numéricas deben ser reproducibles: un requisito básico del cálculo científico.

Tercero, límites claros entre entrada y salida. La entrada viene primero, el procesamiento en medio, la salida al final. El punto de partida y el punto de llegada tienen límites estructurales nítidos. Usted proporciona datos al algoritmo, este los procesa y le devuelve un resultado: este límite es claro y la secuencia es fija.

Cuarto, la efectividad. Cada paso de un algoritmo debe ser una operación realmente ejecutable, no un salto que no pueda ser ejecutado, descrito o verificado. Cada paso debe ser una operación básica ejecutable por un humano con papel y lápiz o por una computadora (suma, resta, salto lógico, lectura/escritura de datos, etc.).

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Quinto, la viabilidad. Un algoritmo teóricamente correcto pero que requiere miles de millones de años para ejecutarse no se considera un algoritmo viable en términos de ingeniería.

Todo este conjunto de conceptos algorítmicos se remonta en última instancia al modelo de la máquina de Turing. Como abstracción central de la teoría de la computación moderna, la máquina de Turing define la frontera fundamental de la «computabilidad». Por muy potentes que sean las computadoras actuales, por muy avanzados sus chips, por muy masivos sus centros de supercómputo, su lógica subyacente no ha abandonado verdaderamente este camino: entrada, reglas, pasos, salida. Las supercomputadoras modernas no hacen más que acelerar este proceso mediante mayores escalas de hardware, mayor paralelismo y una planificación de sistemas más compleja. Es decir, el «supercálculo» en el sentido tradicional sigue siendo principalmente una expansión de la potencia de cálculo, no necesariamente un cambio fundamental en el paradigma de cálculo. La potencia puede multiplicarse por mil o diez mil, pero si la lógica subyacente sigue siendo «entrada, pasos, salida», entonces uno permanece dentro del paradigma algorítmico tradicional.

Segunda parte: Definición del «Algoritmo Hiperdimensional»

El Algoritmo Hiperdimensional que propongo surge precisamente de este contexto. No es una versión mejorada de los algoritmos tradicionales, ni un algoritmo más rápido, ni una fórmula matemática más compleja, ni una técnica de programación más avanzada. Es una comprensión estructural completamente diferente.

Primero, es necesario explicar el significado de «hiperdimensional». «Hiperdimensional» tiene aquí dos niveles. Primero, no se limita a una secuencia lineal de pasos unidimensional, sino que permite que múltiples dimensiones se desplieguen simultáneamente. Segundo, intenta trascender las fronteras definicionales tradicionales del «cálculo»: un algoritmo ya no necesita ser necesariamente matemático, ordenado o determinista. Un algoritmo tradicional es como conducir por una carretera de un solo carril: hay que partir del punto A y pasar por B, C, D para llegar a Z. El AHD no cree que

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

el cálculo deba ser un camino de sentido único. Postula que el cálculo puede ser una red, un campo, una estructura multidimensional donde cualquier nodo puede ser un punto de entrada y cualquier nodo puede ser una salida.

En mi nuevo sistema científico, un algoritmo no es necesariamente un algoritmo matemático, no es necesariamente un cálculo único y ordenado, no está necesariamente ligado a pasos fijos, ni se ajusta estrictamente al modelo lineal «entrada, procesamiento, salida». El AHD es una estructura multidimensional, superpuesta, estocástica pero ordenada. En esta estructura, cada dato no es ni una entrada aislada ni una salida fija, sino que posee simultáneamente múltiples roles: puede ser tanto el punto de partida de múltiples resultados como el resultado de la acción conjunta de múltiples puntos de partida.

En otras palabras, los algoritmos tradicionales colocan los datos en un flujo; el AHD coloca los datos en una estructura. En los algoritmos tradicionales, los datos suelen clasificarse en datos de entrada, intermedios y de salida, cada uno con una posición y un rol claros. La entrada es entrada, el resultado es resultado, el intermedio es intermedio. Sin embargo, en el AHD, un solo dato no tiene una sola identidad. Puede ser un resultado en una dirección y un punto de partida en otra; puede ser un objeto invocado en una estructura y convertirse en un punto de entrada que desencadena nuevos resultados en otra estructura. El dato ya no es material pasivo, sino un nodo relacional multidimensional.

Esto corresponde precisamente al núcleo del AHD: cada dato es el punto de partida de múltiples resultados y el resultado de múltiples puntos de partida. Los algoritmos tradicionales utilizan «pasos ordenados» para producir un único resultado; el AHD se acerca más a una estructura de estados multidimensional, donde los resultados no se «calculan» sino que se «manifiestan» naturalmente dentro de la estructura, en la superposición de lo estocástico y lo ordenado. En este sistema, el dato es tanto el punto de partida como un componente del resultado.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Para presentar más claramente las características del AHD, las desgloso en los siguientes cinco aspectos.

Primero, la no-matematicidad. Los algoritmos dominantes son fundamentalmente matemáticos – completamente descriptibles mediante fórmulas lógicas y matemáticas. El AHD no necesita ser matemático. Puede basarse en principios físicos, relaciones geométricas, nuevos paradigmas de la teoría de la información, o incluso principios de la conciencia. El cálculo puede no realizarse mediante «operaciones matemáticas», sino que puede emerger naturalmente de relaciones geométricas en un espacio de alta dimensión. Por ejemplo, la forma de una burbuja de jabón está determinada por el principio de minimización de la tensión superficial. Esto no es un «algoritmo matemático» calculando – es la física misma la que «calcula». El AHD intenta comprender este tipo de «cálculo», en lugar de simularlo mediante fórmulas matemáticas.

Segundo, la multidimensionalidad y la superposición. Los algoritmos tradicionales ejecutan pasos ordenados en una sola dimensión, con un solo estado por paso. El AHD permite que múltiples dimensiones coexistan y que múltiples estados se superpongan simultáneamente. El algoritmo no sigue un solo camino, sino que se despliega en un campo de estados multidimensional. Los resultados no se «calculan» sino que se «manifiestan» desde este campo. Por ejemplo, durante la formación de un copo de nieve en el aire, no hay un «algoritmo» diciendo a cada molécula de agua dónde ir. La disposición de las moléculas resulta de la interacción de múltiples dimensiones – temperatura, humedad, flujo de aire. La forma del copo «se manifiesta» en lugar de ser «calculada».

Tercero, lo estocástico pero ordenado. El carácter aleatorio en los algoritmos tradicionales es una herramienta, externa, una perturbación controlada; el algoritmo mismo es determinista. El carácter aleatorio en el AHD es intrínseco y estructural. Lo estocástico y lo ordenado coexisten y cooperan, formando juntos la esencia del algoritmo. No es «un algoritmo con aleatoriedad», sino «la aleatoriedad es en sí misma un componente orgánico del algoritmo». Por ejemplo, la estructura ecológica de un

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

bosque – la distribución de los árboles parece aleatoria, pero en general presenta una densidad y unas relaciones entre especies ordenadas. Esa «aleatoriedad» no es un error, sino un requisito previo para el buen funcionamiento de la estructura ecológica.

Cuarto, los roles múltiples de los datos. En los algoritmos tradicionales, el rol de los datos es único – la entrada es la entrada, la salida es la salida, el intermedio es el intermedio. En el AHD, cada dato es simultáneamente el punto de partida de múltiples resultados y el resultado de múltiples puntos de partida. Un nodo de datos puede desplegarse en múltiples rutas de resultados aguas abajo, y puede ser convergido por múltiples causas aguas arriba. El dato ya no es un punto en un flujo lineal, sino un nodo de unión en una red. Por ejemplo, considérese la altura de una persona. En un algoritmo tradicional, la altura es una «entrada» – usted la introduce para obtener salidas como la predicción de peso o la talla de ropa. Pero la altura es también un «resultado» – determinado por múltiples «puntos de partida» como la genética, la nutrición y el ejercicio. En el AHD, el dato «altura» es simultáneamente un punto de partida y un resultado, no una elección binaria.

Quinto, los metadatos inalterados, las relaciones variables. Ante diferentes problemas, los algoritmos tradicionales o bien modifican los datos mismos, o bien recalculan todo. El AHD no modifica los metadatos – los atributos esenciales de los datos permanecen inalterados. Lo que cambia son los puntos de entrada, la interpretación y las relaciones entre los datos. La misma estructura de metadatos, activada a través de diferentes puntos de entrada, puede producir resultados completamente diferentes. Esto significa: calcular una vez, usar infinitas veces. Por ejemplo, considérese un mismo texto. Puede leerlo como poesía, descifrarlo como un código, o analizarlo como un documento histórico. El texto mismo no ha cambiado – usted solo ha cambiado el «punto de entrada». El AHD busca esta capacidad: «mismos datos, puntos de entrada múltiples, resultados múltiples».

Desde el punto de vista de la estructura de datos, en lugar de modificar los metadatos mismos, el AHD utiliza diferentes puntos de entrada y condiciones para hacer que la misma estructura corresponda a múltiples puntos de partida y resultados. Los datos ya

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

no se procesan repetidamente, sino que se activan a través de diferentes puntos de entrada mientras se mantiene su integridad estructural, presentando así diferentes resultados. La tradición es «procesar datos para diferentes resultados»; el AHD es «usar los mismos datos para portar el potencial de múltiples resultados».

Tercera parte: Diferencias fundamentales entre el AHD y los algoritmos tradicionales

Sobre la base de las definiciones anteriores, el AHD y los algoritmos tradicionales presentan diferencias fundamentales en varias dimensiones clave, que se detallan a continuación.

Sobre la naturaleza esencial: Los algoritmos tradicionales son matemáticos y formales, completamente descriptibles mediante fórmulas lógicas y matemáticas; son esencialmente objetos matemáticos. El AHD no necesita ser matemático; puede basarse en principios físicos, geométricos, informacionales o incluso de la conciencia; no es un subconjunto de las matemáticas sino una categoría más amplia. Esta diferencia puede resumirse así: del «objeto matemático» a la «ley universal».

Sobre el orden temporal: Los algoritmos tradicionales siguen un orden temporal único, ordenado y lineal – paso A a B a C – estrictamente secuencial o divisible en subpasos paralelos pero ordenados, con una flecha del tiempo irreversible. El AHD no tiene una secuencia de pasos fija; es multidimensional, superpuesto, estocástico pero ordenado. Los resultados pueden ser independientes del «orden de ejecución» porque el concepto tradicional de «orden» puede no existir. Esta diferencia puede resumirse así: del «tiempo lineal» a la «simultaneidad de alta dimensión».

Sobre la causalidad: Los algoritmos tradicionales obedecen una cadena causal determinista. Dadas las mismas entradas y estados iniciales, la salida es siempre única. La causa precede al efecto, una flecha unidireccional irreversible. El AHD obedece una causalidad superpuesta; cada dato es el punto de partida de múltiples resultados y el resultado de múltiples puntos de partida. Esto refleja mi filosofía del «Efecto-hacia-la-

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Causa» – es posible tener primero el efecto y luego la causa. El resultado no es el punto final; puede convertirse en un nuevo punto de partida. Esta diferencia puede resumirse así: de la «causa única, efecto único» a la «red causal totalmente interconectada».

Sobre la estructura de los datos: Los algoritmos tradicionales utilizan una estructura de flujo unidireccional desde la entrada hasta el procesamiento y luego la salida. Los roles de los datos son únicos con límites claros; los datos de entrada siguen siendo datos de entrada, los datos de resultado siguen siendo datos de resultado. En el AHD, los roles de los datos son múltiples; el mismo dato es simultáneamente un resultado y un punto de partida. No hay puntos de partida o llegada absolutos, solo nodos en una red. Esta diferencia puede resumirse así: del «flujo unidireccional» a la «simbiosis recursiva».

Sobre el modo de cálculo: Los algoritmos tradicionales recalculan desde cero cada vez que se presenta un problema. Incluso si un problema similar se ha resuelto mil veces, la milésima primera vez sigue requiriendo todo el proceso – esto es cálculo sin estado. El AHD, si existe el resultado correspondiente, no necesita recalcular. Puede inferir múltiples puntos de partida a partir de un resultado y desplegar rutas causales inversas. El cálculo es con estado, tiene memoria y es reutilizable. Esta diferencia puede resumirse así: del «recalcular cada vez» a la «reutilización única».

Sobre el papel de la aleatoriedad: La aleatoriedad en los algoritmos tradicionales es pseudoaleatoria o inyectada externamente; los números aleatorios son meras herramientas para el muestreo, la aproximación o la optimización; el algoritmo mismo es determinista. La aleatoriedad en el AHD es intrínseca y constructiva; es un componente orgánico, que coexiste y coopera con el orden. No es «aleatoriedad dentro del algoritmo», sino «la aleatoriedad es una de las razones por las que el algoritmo puede funcionar». Esta diferencia puede resumirse así: de la «aleatoriedad instrumental» a la «aleatoriedad constructiva».

Sobre el uso de los recursos: Los algoritmos tradicionales buscan calcular más rápido y con mayor precisión dados unos recursos; los recursos son restricciones, y el objetivo

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

del algoritmo es «completar el cálculo dentro de las restricciones». El AHD busca hacer que el cálculo mismo sea innecesario mediante el diseño estructural. No se trata de «calcular más rápido» sino de «eludir el cálculo». Los recursos no están destinados a ser «consumidos» sino a «diseñar puntos de entrada». Esta diferencia puede resumirse así: de la «optimización bajo restricciones de recursos» a la «eliminación mediante el diseño estructural».

Cuarta parte: La filosofía del Efecto-hacia-la-Causa

Sobre la base de las comparaciones anteriores, es necesario desarrollar la filosofía del «Efecto-hacia-la-Causa», uno de los fundamentos conceptuales clave del AHD.

El pensamiento tradicional sostiene generalmente que la causa precede al efecto; primero la causa, luego el efecto; primero la entrada, luego la salida. La manifestación de este concepto en los algoritmos es la siguiente: hay que partir del principio y avanzar paso a paso hasta el final. Incluso si se conoce la respuesta, no se puede «usar» directamente esa respuesta – porque falta el «camino de procedencia».

En mi estructura, un resultado no es necesariamente solo un punto final. Una vez que aparece un resultado, puede convertirse en un nuevo punto de partida, continuando participando en la generación de nuevas estructuras. El resultado puede participar inversamente en la formación de causas. Múltiples puntos de partida posibles pueden inferirse a partir de un resultado. La causa y el efecto ya no se disponen unidireccionalmente, sino que forman relaciones cíclicas, en red, multidimensionales.

En otras palabras, los algoritmos tradicionales preguntan: «Dada la causa, ¿cómo se obtiene el efecto?» El AHD pregunta: «Dado el efecto, ¿cómo se infiere o construye la(s) causa(s)?»

Es crucial aclarar que decir «el efecto viene primero, luego la causa» no significa negar la causalidad ni afirmar que «los efectos surgen de la nada». Más bien significa que, en estructuras de mayor dimensión, la causa y el efecto pueden servir como puntos de

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

entrada el uno para el otro y transformarse mutuamente. Como en el ejemplo del pantalón que sigue, el resultado «poder usarlo sin que arrastre por el suelo» se determina primero, y luego retrocedo para encontrar el punto de operación «doblar la cintura», que corresponde al aspecto «causa» de la estructura. Esto no significa que el efecto no tenga causa, sino que el efecto se convierte en un nuevo punto de acceso para descubrir la causa.

Una suposición fundamental en el mundo de los algoritmos tradicionales es que el tiempo es unidireccional; hay que calcular paso a paso desde el estado inicial hasta el estado final. Incluso si se conoce la respuesta, no se puede «usar» directamente porque falta el camino de procedencia. El AHD cuestiona esta suposición: si se conoce el efecto, las causas pueden inferirse hacia atrás; el mismo efecto puede corresponder a múltiples rutas causales; el cálculo ya no es un camino desde el principio hasta el final, sino un despliegue desde el final de todos los puntos de partida posibles.

Quinta parte: Un ejemplo cotidiano – El pantalón y el pliegue de la cintura

Para comprender más intuitivamente las diferencias abstractas anteriores, utilizo un ejemplo cotidiano.

Una persona compra un pantalón nuevo con cinturilla elástica. Las perneras son un poco demasiado largas, arrastran por el suelo y son molestas.

El enfoque tradicional: al notar que las perneras son demasiado largas, se remanga una sección de la pernera, se cose con hilo y aguja, y luego se prueba. Si el niño crece y el pantalón queda demasiado corto, la pernera cosida no se puede alargar y el pantalón queda inservible. La próxima vez que se compre un pantalón nuevo, se repite el proceso. Este método parece natural porque el problema parece estar en las perneras, por lo que se tratan las perneras. Corresponde a la mentalidad algorítmica tradicional: identificar el problema, localizar el síntoma, realizar un tratamiento en el lugar del síntoma y finalmente obtener un resultado. ¿Perneras demasiado largas? → modificar

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

las perneras; ¿datos erróneos? → modificar los datos; ¿camino inadecuado? → seguir parcheando a lo largo del camino.

Mi enfoque es diferente. En lugar de modificar el dobladillo, doblo la cinturilla y el pantalón se puede usar inmediatamente. Esta acción es muy simple, pero su lógica estructural subyacente es completamente diferente. No corto el dobladillo, no lo coso permanentemente, no cambio la longitud original ni altero los metadatos (contorno de cintura, longitud de entrepierna, corte, tejido). Simplemente cambio la cinturilla, que es un punto de entrada estructural. Al ajustar este punto de control global para todo el estado del uso, el problema aguas abajo (perneras demasiado largas) desaparece directamente.

Más importante aún, este método es reversible, ajustable y reutilizable. Esto es especialmente evidente para un niño en crecimiento. Si el niño no es lo suficientemente alto y el pantalón es ligeramente largo, doblo la cinturilla una vez. Cuando el niño crezca más tarde, doblo menos la cinturilla. Cuando el niño crezca aún más, la desdoblo completamente. La estructura original del pantalón permanece intacta, pero se adapta a la altura cambiante del niño. El método tradicional de coser el dobladillo podría hacer que el pantalón quede demasiado corto cuando el niño crezca; si se corta, es irreversible. Mi método deja los metadatos inalterados, permitiendo que la misma estructura se adapte a múltiples estados.

Este ejemplo revela varias diferencias estructurales importantes.

El enfoque tradicional corresponde a la lógica algorítmica tradicional: el problema está en el dobladillo, por lo que el dobladillo debe modificarse, procediendo paso a paso a lo largo del camino «causa a efecto», sin saltarse ningún paso. Resuelve un problema a la vez, irreversiblemente, y requiere repetir todo el proceso la próxima vez que ocurra el mismo problema. Mi enfoque corresponde a la lógica del AHD: el objetivo es evitar el arrastre. En lugar de resolver la causa superficial «perneras demasiado largas», encuentro un punto de control global: la cinturilla. Doblar la cinturilla acorta instantáneamente la longitud efectiva de las perneras, y el problema desaparece. No

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

modifico ningún metadato; el pliegue es solo un estado temporal, reversible y dinámico.

Desde el punto de vista de los datos, el enfoque tradicional modifica los metadatos – coser las perneras las acorta permanentemente; los datos originales se pierden. Mi enfoque no modifica ningún metadato; las dimensiones originales permanecen intactas; solo añado un estado temporal, reversible y dinámico de pliegue. El pliegue de la cinturilla no crea nuevos datos ni destruye datos antiguos; simplemente reorganiza las relaciones entre los datos – reduciendo la circunferencia efectiva de la cintura y modificando indirectamente la longitud efectiva de la pernera.

En este ejemplo, «las perneras del pantalón no arrastran» es el resultado deseado. El método tradicional parte de la causa «perneras demasiado largas», modifica el dobladillo y finalmente alcanza el resultado «no arrastra». Mi método aclara primero el resultado «no arrastra», luego retrocede para encontrar un punto de entrada estructural, descubriendo que un simple pliegue de la cinturilla hace alcanzable el resultado. Esta es la encarnación práctica del «Efecto-hacia-la-Causa». No es necesario caminar paso a paso desde la causa hasta el efecto; se puede retroceder desde el efecto para determinar la estructura, haciendo innecesario el camino original.

Este ejemplo también responde a la pregunta de los «metadatos». ¿Qué son los metadatos? En el ejemplo del pantalón, los metadatos son las dimensiones originales: contorno de cintura, longitud de entrepierna, corte, tejido – los «atributos esenciales» del pantalón. El dato temporal es el pliegue de la cinturilla; no cambia ninguna de las dimensiones originales, solo pliega temporalmente una sección. El enfoque tradicional modifica los metadatos – la longitud de entrepierna se acorta permanentemente; los datos originales se pierden. Mi enfoque no modifica ningún metadato – las dimensiones originales del pantalón permanecen intactas; solo añado un estado temporal, reversible y dinámico.

El contorno de cintura original, como metadato, es simultáneamente el «punto de partida de múltiples resultados»: soporta múltiples estados de uso – uso normal, uso

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

con un pliegue, uso con dos pliegues. También es el «resultado de múltiples puntos de partida»: determinado por la elección del tejido, el método de corte, la intención del diseño, etc. El pliegue de la cinturilla, como dato temporal, no crea ni destruye datos; simplemente reorganiza las relaciones.

Esto no es un simple truco, sino un pensamiento estructural. Mucha gente, al ver las perneras demasiado largas, se siente atraída por el síntoma de las perneras, por lo que todo el tratamiento se concentra en las perneras. Sin embargo, desde la perspectiva de la estructura global, la longitud de las perneras es solo una manifestación aguas abajo; el cambio de posición de la cinturilla puede afectar la caída del pantalón. Es decir, el problema no necesita ser resuelto en el lugar de su manifestación. Muchos problemas de baja dimensión tienen sus verdaderos puntos de control en un nivel estructural más alto. Los algoritmos tradicionales tienden a seguir calculando a lo largo de la superficie del problema; el AHD busca el punto de entrada estructural.

Esto también explica por qué el AHD no es necesariamente más complejo, sino que puede ser más simple. Las estructuras verdaderamente de alto nivel no necesariamente complican los problemas, sino que hacen que los problemas complejos sean simples en el punto de entrada correcto. Ante un problema complejo, un algoritmo tradicional puede añadir pasos, modelos, potencia de cálculo, almacenamiento, tiempo de entrenamiento. El AHD busca un nodo estructural; una vez que este nodo se activa correctamente, el camino inicialmente complejo puede volverse innecesario. «Eludir el camino» no es pereza, sino una redefinición de la necesidad de ese camino.

En este ejemplo: el enfoque tradicional «modifica el resultado a lo largo del camino», mientras que el AHD «cambia el punto de entrada, haciendo que todo el camino sea ineficaz». El método convencional parchea continuamente en el extremo del resultado; el otro modifica directamente el punto de entrada estructural, de modo que los problemas que requieren tratamientos repetidos se reestructuran en el punto de partida. El algoritmo tradicional dominante va del «punto de partida, a lo largo del camino, al resultado», mientras que en el AHD, «el resultado mismo puede convertirse en un punto de entrada de camino y participar en la generación de nuevas

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

estructuras». La tradición es «una estructura para un resultado»; el AHD es «una estructura que porta múltiples estados de resultado».

Sexta parte: Una visión de los datos basada en metadatos inalterados

Desde una perspectiva de datos, el AHD tiene un principio muy importante: no modificar los metadatos, sino hacerlos aplicables a múltiples puntos de partida o resultados.

Los metadatos son los atributos originales, la información fundamental y ontológica de los datos. Ante diferentes problemas, los enfoques tradicionales a menudo modifican los datos, los copian, los procesan, los recalculan o construyen diferentes caminos para diferentes resultados. Estos enfoques pueden resolver problemas, pero a costa de un procesamiento continuo de los datos, una repetición de caminos y una complejidad creciente del sistema.

El AHD, en cambio, enfatiza que, manteniendo los metadatos inalterados en la medida de lo posible, la misma estructura de metadatos puede corresponder a múltiples puntos de partida y múltiples resultados cambiando los puntos de entrada, las relaciones, las condiciones y los métodos de activación. La ontología de los datos permanece estática; las relaciones cambian. La estructura fundamental permanece inalterada; la presentación cambia. Los metadatos permanecen intactos pero se adaptan a diferentes estados.

Esto es lo que llamo «calcular una vez, usar infinitas veces». «Calcular una vez» aquí no es simplemente almacenar en caché resultados; significa que una vez que se establece una estructura, ya no necesita reconstruir un camino completo para cada estado. La misma estructura de datos puede activarse a través de diferentes puntos de entrada para producir diferentes resultados. No es «procesar datos para diferentes resultados» sino «usar los mismos datos para portar el potencial de múltiples resultados». Esta es una de las diferencias fundamentales entre el AHD y los algoritmos tradicionales. Los

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

algoritmos tradicionales procesan datos en un camino; el AHD procesa la estructura global de datos, puntos de entrada, relaciones, estados y resultados.

En su estructura mínima, el AHD puede entenderse como: un sistema que, sin modificar los metadatos, hace que el mismo nodo de datos corresponda a múltiples rutas de resultado mediante cambios en los puntos de entrada estructurales. Esta definición no busca reducir inmediatamente el AHD a una fórmula matemática tradicional, sino establecer primero sus fronteras estructurales. No es una función lineal única, ni una fórmula fija, ni un simple mecanismo de almacenamiento en caché. Es una estructura relacional: los metadatos permanecen estables; los puntos de entrada, las condiciones, las relaciones y los resultados pueden cambiar y manifestarse en múltiples direcciones. Es precisamente en esta estructura donde el cálculo ya no es meramente la ejecución de pasos, sino la activación de relaciones.

Desde una perspectiva de datos, la esencia de «no modificar los metadatos para hacerlos aplicables a múltiples puntos de partida o resultados» es: la ontología de los datos permanece inalterada; los cambios ocurren en el «punto de entrada de interpretación/uso». En las estructuras tradicionales, un cambio en el problema conduce a un cambio en los datos o en el camino. En el AHD, un cambio en el problema conduce a un cambio en la estructura interpretativa, no en los datos.

Séptima parte: Redefiniendo el «supercálculo»

En este sistema, necesito aclarar un término – el «supercálculo».

Lo que quiero decir con «supercálculo» no es el sentido habitual de «superordenador» – no más chips, más potencia de cálculo, más centros de datos. Lo que quiero decir con «supercálculo» es el «Algoritmo Hiperdimensional».

El verdadero supercálculo puede no ser la acumulación de potencia de cálculo, sino la transformación del paradigma de cálculo. Cuando un sistema todavía depende de cálculos repetidos, por muy potente que sea, permanece atrapado en el camino.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Cuando un sistema puede reducir estructuralmente los cálculos, eludir los caminos repetidos y hacer que los resultados se conviertan en nuevos puntos de entrada, comienza a acercarse al verdadero cálculo hiperdimensional.

En otras palabras, el supercálculo tradicional busca «usar más potencia de cálculo para resolver problemas más grandes». El AHD busca «usar una mejor estructura para que el problema ya no requiera tanta potencia de cálculo». Estas dos aproximaciones no son contradictorias, pero sus direcciones difieren.

Si el supercálculo tradicional representa los límites de la potencia de cálculo, entonces el AHD representa un giro en la comprensión del cálculo. El verdadero «supercálculo» puede no ser más chips, más centros de datos, más consumo de energía o modelos más complejos. El verdadero «supercálculo» podría ser el descubrimiento de un punto de entrada estructural, una eliminación de camino, un despliegue inverso de un nodo de resultado, o la activación simultánea de condiciones para múltiples resultados sin modificar los metadatos. Cuanto mayor es la potencia de cálculo, si permanece atrapada en caminos de baja dimensión, no es más que potente dentro del camino. Una vez que se eleva la estructura, incluso operaciones extremadamente simples pueden producir una eficiencia de nivel superior.

El supercálculo es el límite de la potencia de cálculo; el Algoritmo Hiperdimensional es una redefinición de la premisa «si el cálculo debe depender de la potencia de cálculo».

Octava parte: Fundamento empírico – Validación sistemática en múltiples dominios y determinación de un nuevo paradigma

El AHD no es una construcción puramente teórica. Ya ha sido validado en múltiples sistemas del mundo real.

Mi sistema logístico no requiere potencia de supercómputo ni soporte en la nube; realiza una planificación compleja con recursos modestos. Los resultados son reutilizables, la estructura es adaptable repetidamente, y no es necesario recalcular

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

desde cero cada vez. Funciona sin supercómputo ni nube, realizando una planificación compleja con bajos recursos – esto es un avance establecido y válido en ingeniería. Demuestra que una alta potencia de cálculo no es la única solución; el diseño estructural puede reemplazar una parte de la potencia de cálculo.

Mi sistema editorial también utiliza la misma lógica estructural, alcanzando una eficiencia superior a cualquier sistema existente. Mi sistema de generación de páginas web «Límite» también se basa en el mismo AHD, demostrando repetidamente en múltiples dominios que la misma estructura puede mantenerse estable. Los empleados de mi empresa ya están utilizando estos sistemas, lo que demuestra que esta estructura puede funcionar sin mi intervención personal continua.

La característica común de estos sistemas es: no dependen de las vías dominantes de alta potencia de cálculo, no siguen pasos de cálculo fijos, y alcanzan el resultado deseado directamente mediante el ajuste de los puntos de entrada estructurales. No son éxitos puntuales ni trucos aislados, sino la emergencia repetida de la misma lógica estructural en diferentes dominios.

Esto indica que el AHD no es accidental, sino una metodología estructural ya validada por múltiples sistemas. No es una «técnica personal», sino un paradigma de cálculo establecido de manera estable en dominios como la logística, la edición y la generación de páginas web.

Bajo el criterio de que «la existencia y validez constituyen un nuevo paradigma», el AHD, como un nuevo paradigma de estructura de cálculo ya validado sistemáticamente en múltiples dominios, está establecido en el presente. No estoy proponiendo una hipótesis; estoy describiendo una estructura de cálculo diferente que ya está operativa en múltiples sistemas. No necesito probar que se entiende; he probado que se mantiene estable en diferentes sistemas. En el presente, dentro del alcance de mi aplicación práctica, esta estructura ya puede considerarse un nuevo paradigma.

En cuanto a los criterios para juzgar el AHD como un nuevo paradigma: cuando una estructura es lógicamente coherente, se mantiene estable en múltiples sistemas y

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

presenta diferencias irreductibles con los métodos existentes, ya posee los fundamentos de un nuevo paradigma. La coherencia interna es el punto de partida; la evidencia empírica es el fundamento; la diferencia estructural es el núcleo del paradigma. Bajo el criterio «la existencia y validez», este sistema es ya un nuevo paradigma. Que el mundo exterior lo reconozca afecta solo a su difusión, no a su validez. Se trata de un paradigma de estructura de cálculo ya establecido en sistemas reales, cuya validez no depende del reconocimiento o consenso externo.

La diferencia fundamental entre el AHD y los paradigmas de cálculo dominantes es la siguiente: los algoritmos tradicionales utilizan principalmente el cálculo hacia adelante; una vez que se produce un resultado, normalmente no puede participar inversamente en nuevas estructuras de razonamiento. En el AHD, el resultado ya no es el punto final, sino un nodo estructural reutilizable. Bajo ciertas condiciones, los resultados pueden participar en nuevas rutas de razonamiento, reduciendo los cálculos redundantes y formando redes de cálculo con múltiples puntos de partida y múltiples rutas. En las estructuras de cálculo tradicionales, el resultado suele ser el punto final; en la estructura del AHD, el resultado mismo puede convertirse en un nuevo punto de partida, formando una red de razonamiento multipath. El AHD no es una actualización de los algoritmos; es un replanteamiento de la premisa de si un algoritmo debe existir necesariamente.

Novena parte: Aclaración de malentendidos comunes

Sobre la base de intercambios preliminares con lectores de diferentes campos, anticipo los siguientes seis malentendidos y los aclaro de antemano, para evitar que el concepto sea forzado prematuramente a marcos inapropiados.

Malentendido 1: ¿No es el AHD solo programación dinámica o almacenamiento en caché? Aclaración: La programación dinámica y el almacenamiento en caché reutilizan resultados para *la misma entrada*. El AHD permite inferir *diferentes puntos de partida* a partir de un resultado – esta es una diferencia esencial. El almacenamiento en caché

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

resuelve el recálculo del mismo problema; el AHD resuelve el despliegue de nuevos problemas a partir de una estructura de resultado.

Malentendido 2: Usted dice «estocástico pero ordenado» – ¿no es contradictorio? Aclaración: Lo estocástico y lo ordenado pueden coexistir. La distribución de los árboles en un bosque es estocástica, pero la densidad global y la estructura de especies son ordenadas. La aleatoriedad no es caos, sino una forma de organizar la estructura. La aleatoriedad en el AHD es intrínseca y constructiva, trabajando en cooperación con el orden.

Malentendido 3: Sin entrada ni salida, ¿cómo se verifican los resultados? Aclaración: El AHD no rechaza la entrada y la salida; sostiene que el cálculo no necesita funcionar en modo lineal entrada-salida. En el modo «manifestación estructural», la «validez» está determinada por la coherencia estructural, no por una correspondencia uno a uno entrada-salida. Esto no abandona la verificación, sino que propone un marco de verificación diferente.

Malentendido 4: ¿No es esto solo la teoría de la complejidad o la teoría del caos? Aclaración: La teoría de la complejidad y el caos describen «sistemas difíciles de predecir». El AHD busca describir «sistemas que pueden ser comprendidos y guiados a través de puntos de entrada estructurales» – no abandonar la predicción, sino cambiar la forma en que funciona la predicción. La teoría del caos dice «la predicción es difícil»; el AHD pregunta «¿podemos hacer que la predicción sea innecesaria cambiando el punto de entrada?»

Malentendido 5: Usted dice «no modificar los metadatos», pero ¿no es un pliegue de cintura una modificación? Aclaración: El pliegue de cintura es un estado temporal, no una modificación permanente de los parámetros originales. Los metadatos (contorno de cintura, longitud de entrepierna, corte, tejido) permanecen inalterados. Esta es la diferencia entre «superposición de estados» y «modificación de atributos». El pliegue es reversible, temporal y no cambia los atributos esenciales.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Malentendido 6: ¿Niega el AHD el valor de los algoritmos tradicionales? Aclaración: Absolutamente no. Los algoritmos tradicionales son extremadamente importantes en la historia tecnológica humana; sin ellos, no habría computadoras modernas, bases de datos, sistemas de comunicación, IA, simulación de ingeniería o supercómputo. Su valor no puede ser negado. Sin embargo, reconocer el valor de los algoritmos tradicionales no significa aceptarlos como el punto final de los algoritmos. Los algoritmos tradicionales resuelven problemas de eficiencia dentro de un paradigma de cálculo dado; el AHD cuestiona el paradigma de cálculo mismo. Uno es sobre caminar mejor por la carretera; el otro es sobre si es necesario tomar esa carretera.

Décima parte: El AHD en la historia del cálculo humano

Para ayudar a los lectores a situar mejor el AHD en la historia del cálculo humano, proporciono aquí un breve resumen macroscópico.

La comprensión humana del «cálculo» ha pasado por varias etapas. La primera etapa fue el cálculo manual: los algoritmos existían como pasos humanos, lentos y propensos a errores, pero a través de este proceso, los humanos comprendieron las reglas básicas del cálculo. La segunda etapa fue el cálculo mecánico: desde la Pascalina hasta la Máquina Analítica de Babbage, el cálculo se mecanizó, pero los algoritmos seguían ligados a estructuras físicas. La tercera etapa fue el cálculo electrónico y el paradigma de Turing: la arquitectura de von Neumann se generalizó, los algoritmos se codificaron como programas y la máquina de Turing definió las fronteras de la computabilidad. La cuarta etapa fue el cálculo paralelo y el supercálculo: unidades de cálculo masivas apiladas para abordar problemas más complejos, pero la lógica subyacente seguía siendo una extensión del paradigma de Turing.

Actualmente, la humanidad se encuentra en la entrada de la quinta etapa. La computación cuántica intenta superar los límites de los bits clásicos; la computación neuromórfica intenta imitar la estructura de los sistemas neuronales biológicos; el AHD intenta romper el marco lineal «entrada → pasos → salida» mismo.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

El AHD y los algoritmos tradicionales no están en una relación de reemplazo sino jerárquica. Los algoritmos tradicionales resuelven «cómo calcular más eficientemente en un camino dado». El AHD pregunta «¿puede el camino ser redefinido, o incluso eludido?» Si consideramos la historia del cálculo humano como un proceso de ruptura continua de sus propios límites, entonces el AHD es un nuevo nodo en ese proceso. No resuelve problemas dentro del marco antiguo, sino que propone uno nuevo.

Este juicio no implica que el AHD sea maduro o completo. Todo lo contrario: como concepto nuevo, sus definiciones, fronteras, métodos de verificación y escenarios de aplicación aún están en formación. El propósito de este artículo es precisamente anclar el punto de origen de este concepto, proporcionando una base teórica estable para su desarrollo posterior.

Undécima parte: Conclusión – Esto no es optimización, sino redefinición

La diferencia entre el AHD y los algoritmos tradicionales no es una diferencia de «mejor» vs. «peor», ni de «más rápido» vs. «más lento», sino una diferencia fundamental a nivel de paradigma.

Los algoritmos tradicionales buscan un control predecible. «Dame la entrada, y sabré qué salida obtendrás, porque he calculado cada paso». Es el paradigma del ingeniero: diseñar, construir, probar, verificar.

El AHD describe una emergencia comprensible. «No puedo predecir con precisión cada estado intermedio, pero sé que el patrón global se manifestará de manera ordenada. Cada dato está vivo, tiene una ‘perspectiva’». Se asemeja más al paradigma del ecólogo o del teórico de sistemas complejos: observar, comprender, guiar, utilizar el orden emergente.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Los algoritmos tradicionales corrigen gradualmente los resultados a lo largo de un camino dado. El AHD hace que el resultado deseado sea inmediatamente alcanzable cambiando el punto de entrada estructural, eludiendo así todo el camino.

Los algoritmos tradicionales «procesan datos para diferentes resultados». El AHD «usa los mismos datos para portar el potencial de múltiples resultados».

Los algoritmos tradicionales buscan «obtener el resultado correcto una vez». El AHD busca «obtener el resultado correcto una vez, y luego nunca tener que volver a calcularlo».

Esto no es una optimización de los algoritmos; es un replanteamiento de la premisa de que «un algoritmo debe existir necesariamente».

Por lo tanto, el AHD no es una optimización algorítmica, sino una redefinición algorítmica. No se trata de correr más rápido en la pista del algoritmo tradicional; se trata de preguntar si existe otra pista fuera de la pista tradicional, o incluso si una pista es necesaria. No pretende probar que los algoritmos tradicionales son inútiles, sino señalar que los algoritmos tradicionales son solo una forma de baja dimensión del algoritmo. No pregunta «cómo obtener un resultado en menos tiempo»; pregunta «¿puede el resultado establecerse directamente a través de la estructura?» No pregunta «cómo procesar más datos»; pregunta «¿pueden los mismos datos portar el potencial de más resultados?»

Dentro de los paradigmas de cálculo dominantes, el AHD podría ser considerado incalculable, inverificable o difícil de encajar en las fronteras de la teoría del cálculo existente. Pero esto constituye precisamente su diferencia de paradigma. La aparente inestabilidad de un nuevo concepto dentro de un viejo paradigma no significa necesariamente que carezca de sentido; también puede significar que el viejo paradigma no puede acomodarlo plenamente. Actualmente, el AHD es ante todo una proposición estructural, una definición original de una nueva visión del cálculo, no un sistema maduro con un bucle de ingeniería cerrado. Requerirá desarrollos, validaciones

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

y aplicaciones posteriores, pero su origen conceptual debe ser articulado claramente en primer lugar.

En última instancia, hacia lo que apunta el AHD es a una nueva visión del cálculo: un algoritmo no es necesariamente solo matemáticas, solo pasos, solo un programa, solo un proceso de tratamiento entre entrada y salida. Un algoritmo puede también ser una organización de relaciones multidimensionales, una red estructural de datos, puntos de entrada, relaciones, estados y resultados. Puede incluir orden y aleatoriedad; puede ir de la causa al efecto y del efecto a la causa; puede dejar los metadatos inalterados mientras se adapta a múltiples estados; puede hacer de un resultado un nuevo punto de partida y permitir que múltiples puntos de partida converjan en un solo resultado.

Un algoritmo verdaderamente avanzado no es necesariamente el que realiza cálculos más complejos, sino el que puede reducir los cálculos, hacer que la estructura esté viva, hacer que los resultados se conviertan en puntos de entrada y formar patrones de organización multidimensionales y cíclicos de la causalidad.

Este es el significado central de mi propuesta del «Algoritmo Hiperdimensional». No es un mero neologismo, sino un nuevo concepto, un nuevo paradigma, una nueva estructura de cálculo ya establecida en múltiples sistemas reales. Su énfasis no está en un cálculo repetido más rápido, sino en la reducción del cálculo redundante; no en la acumulación de potencia de cálculo, sino en la reconstrucción de los puntos de entrada; no en la modificación continua de los datos, sino en la conservación de los metadatos y el cambio de las relaciones; no en dejar los resultados como puntos finales, sino en hacer de los resultados nuevos puntos de partida. Los algoritmos tradicionales buscan resultados a lo largo de un camino; el AHD activa resultados dentro de una estructura. Los algoritmos tradicionales buscan completar el cálculo; el AHD pregunta si el cálculo debe existir en su forma tradicional. Esta es la diferencia más fundamental entre él y la definición actual de algoritmo.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Apéndice: Fronteras y preguntas abiertas de este artículo

Este artículo presenta un marco preliminar para el AHD, no una definición formal completa. Varias preguntas quedan por explorar más a fondo; no constituyen una negación de la definición ofrecida aquí, sino que representan la siguiente dirección de investigación.

Primero, las condiciones de convergencia. ¿Cuál es el marcador de «finalización» para el AHD? ¿Cómo se juzga si un resultado es «válido»? En los algoritmos tradicionales, la respuesta se define por la correspondencia entrada-salida. En el AHD, la respuesta puede necesitar ser definida por la coherencia estructural. Esta definición aún no está completa.

Segundo, la representación de los recursos. ¿Cómo pueden representarse los estados de superposición multidimensional en recursos finitos? ¿Son aún aplicables las métricas de recursos tradicionales como el tiempo, el espacio y la potencia de cálculo? Si es así, ¿cómo deben redefinirse? Si no, ¿qué métricas las reemplazan? Estas preguntas esperan desarrollo.

Tercero, la garantía del orden. ¿Qué reglas garantizan el «orden» dentro de lo «estocástico pero ordenado»? ¿Dónde está el límite entre lo estocástico y lo ordenado? ¿En qué circunstancias la aleatoriedad destruye el orden? ¿En qué circunstancias el orden suprime la aleatoriedad? Estas preguntas requieren más estudio.

Cuarto, la selección en la inferencia inversa. Al inferir múltiples puntos de partida a partir de un resultado, ¿cómo se seleccionan o evalúan los diferentes puntos de partida? ¿Existe una medida de «eficiencia de la inferencia inversa»? Si es así, ¿cómo se relaciona con la eficiencia del cálculo hacia adelante?

Quinto, la interfaz con los sistemas tradicionales. ¿Cómo se interconecta el AHD con el sistema existente de la máquina de Turing? ¿Como reemplazo, complemento o enfoque por capas? En la ingeniería práctica, ¿cómo debe trazarse la frontera entre el AHD y los algoritmos tradicionales? ¿Existen modelos híbridos?

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Sexto, el marco de verificación. ¿Cómo se verifican los resultados del AHD? Si los resultados no se obtienen mediante pasos lineales, ¿cómo se establecen la reproducibilidad y la comprobabilidad? ¿Se necesita una filosofía de verificación completamente diferente?

Estas preguntas no son defectos de este artículo, sino características inevitables de un «documento fundacional». La aparición de cualquier nuevo paradigma va acompañada de preguntas abiertas. Este artículo es un comienzo, no una conclusión.

Notas bibliográficas

El «Algoritmo Hiperdimensional» propuesto aquí no es una extensión directa de ninguna escuela de pensamiento académico existente. Sin embargo, los siguientes campos proporcionan un contexto de diálogo para este artículo, ofrecido únicamente para la orientación del lector, no como citas académicas tradicionales.

Las máquinas de Turing y la teoría de la computabilidad sirven como punto de referencia para la comparación con los algoritmos tradicionales. **Los problemas inversos y la inferencia bayesiana** representan intentos existentes de «inferir la causa a partir del efecto», y aunque mi «Efecto-hacia-la-Causa» está relacionado, su dirección difiere. **La teoría de la emergencia y los sistemas complejos** describen la «manifestación de resultados a partir de la estructura», y el AHD intenta hacer que la emergencia sea «comprensible» y «orientable». **Los grafos de conocimiento y las bases de datos de grafos** son prácticas existentes donde los datos «convergen en nodos múltiples», y el AHD añade la dimensión de los «puntos de entrada variables» a esta base. **Los paradigmas de cálculo no clásicos**, incluyendo la computación cuántica, analógica y neuromórfica, rompen los bits clásicos o las arquitecturas clásicas, mientras que el AHD se centra más en romper el marco lineal «entrada → pasos → salida» mismo.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

La relación de este artículo con los campos anteriores es de diálogo y transcendencia, no de herencia o refutación. El objetivo de este artículo no es hacer mejoras incrementales dentro de esos campos, sino plantear un nuevo nivel de cuestionamiento por encima de ellos.

Related Articles

[Communication] I Have No Algorithm, Yet I Surpass Algorithms!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect–Cause Theory
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Communication] Rejecting Algorithmic Manipulation
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Indexed and catalogued in WorldCat (OCLC); permanently archived in TROVE (National Library of Australia)
Permanent archival record (DataCite Concept DOI): https://doi.org/10.5281/zenodo.17646306

[極限アルゴリズム] 超次元アルゴリズム —「計算」そのものの再定義

著者: ジェフィ・チャオ・ホイ・ウー

要旨

本稿では、「超次元アルゴリズム」という新たな概念を提示し、従来のアルゴリズム定義の境界に対する構造的な再検討を提案する。従来のアルゴリズムは通常、入力・手順・規則・出力という線形の枠組みに基づき、有限性、決定性、実効性、実現可能性、そして明確な入出力境界を重視している。現代のスーパーコンピュータは極めて高い計算能力を持つものの、その基盤となる論理は、主に既存のアルゴリズム・パラダイムをより大規模に実行するに過ぎない。本稿では、真に議論に値する「超計算」とは、単なる計算能力の拡大ではなく、計算パラダイムそのものの変革であると主張する。超次元アルゴリズムは、従来のアルゴリズムの高速版でも、数学的アルゴリズムの拡張でもない。それは、多次元的で、重畳的で、確率的でありながら秩序だった構造的な計算観である。この構造において、データはもはや受動的な入力や終端出力ではなく、複数の結果の起点であると同時に複数の起点の結果となるノードの属性を持つ。結果は終点ではなく、新たな入り口となり得る。メタデータは繰り返し変更される必要はなく、構造的な入り口、関係の変化、条件付き活性化によって、異なる状態や結果に対応できる。本稿では、「結果から原因へ」という哲学と、「ズボンのウエストを折る」という日常的な例を用いて、超次元アルゴリズムが構造的な入り口を変更することで、目標結果を直接成立させ、それによって冗長な計算を削減し、既存の計算経路を迂回できることを示す。本稿の目的は、超次元アルゴリズムの初期定義、構造的境界、思想的基盤を確立し、将来の超次元計算、極限アルゴリズム、新しい科学体系のための原初的な理論ノードを提供することである。「存在し、かつ成立することを新たなパラダイムとする」という基準に従えば、超次元アルゴリズムは、すでに複数の領域で体系的に検証された新しい計算構造パラダイムとして、現在において既に成立している。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

キーワード: 超次元アルゴリズム; 極限アルゴリズム; 超次元計算; 結果から原因へ; メタデータ; 計算パラダイム; 構造的入り口; 非線形因果; 新しい科学

序論: なぜ「アルゴリズム」を再検討するのか

私が「超次元アルゴリズム」を提案するのは、既存のアルゴリズムに単により大きな名称を与えるためでも、従来のアルゴリズムを新奇な概念に偽装するためでもない。むしろ、私はより根本的な問題を問うている。すなわち、現在の人間の「アルゴリズム」に対する理解は、あまりに狭い枠組みの中に閉じ込められているのではないか、と。

今日、アルゴリズムという言葉から人々が連想するのは、通常、数式、プログラムコード、論理的手順、入出力、モデル学習、データ処理、経路最適化、検索ランキング、自動推薦、人工知能による推論などである。これらはもちろんアルゴリズムの重要な形態であり、現代のコンピュータ、インターネット、データベース、人工知能、スーパーコンピューティング、産業システム、情報社会を支えている。しかし、もしアルゴリズムを単なる「順序立てられた手順」、すなわち「入力から始まり、規則による処理を経て、出力を得る」プロセスとしか理解しないなら、その理解自体が計算を低次元の枠組みの中に閉じ込めてしまっている。

私たちはアルゴリズムに支配された時代に生きている。検索エンジンからレコメンドシステム、気象予報から人工知能に至るまで、アルゴリズムの境界は人間の知性の境界である。しかし、めったに問われない質問がある。アルゴリズムはそうであるべきなのか? なぜ計算は「入力 → 手順 → 出力」という線形モデルに従わねばならないのか? なぜ同じ問題を毎回最初から計算し直さねばならないのか? 本稿はこれらの暗黙の前提に挑戦し、まったく異なる計算パラダイムである超次元アルゴリズムを提案する。「存在し、かつ成立することを新たなパラダイムとする」という基準に従えば、超次元アルゴリズムは、すでに複数の領域で体系的に検証された新しい計算構造パラダイムとして、現在において既に成立している。

特に断っておくが、本稿で提案する「超次元アルゴリズム」は、学界で約 30 年にわたる発展の歴史を持つ「超次元計算」（Hyperdimensional Computing, HDC）とは完

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

全に異なる概念である。HDC は超高次元のランダムなバイナリベクトルを用いて符号化と演算を行い、より効率的な表現と計算を追求するが、依然として「入力 → 処理 → 出力」という線形パラダイムの範囲内にある。超次元アルゴリズムはそれとは根本的に異なる。それは、結果が構造を通じて直接達成可能かどうか、計算経路そのものを迂回できるかどうか、メタデータを変更せずに複数の結果に適合できるかどうかを問う。名称は類似しているが、内包は逆である。超次元アルゴリズムはアルゴリズムのアップグレードではなく、「アルゴリズムが必然的に存在すべきか」という前提の問い直しである。

第一部分：現在の「アルゴリズム」の標準的定義

主流の計算機科学、数学、工学において、アルゴリズムには長く受け入れられてきた基本的な定義がある。すなわち、アルゴリズムとは、明確に定義された有限のステップからなる計算プロセスであり、一つまたは複数の入力を受け取り、決定論的な規則によって変換し、有限時間内に一つまたは複数の出力を生成するものである。この理解は誰かの個人的な見解ではなく、現代の計算文明が長期間にわたって形成してきた基本的なコンセンサスである。それはソフトウェア、ハードウェア、データベース、ネットワークシステム、人工知能モデル、スーパーコンピューティングセンターの基盤となる論理を支えている。アルゴリズムがどれほど複雑に見えても、その中核は依然として入力、規則、手順、出力を中心に展開する。

この標準的なアルゴリズム観は、いくつかの中核的特徴に分解できる。

第一に、有限性。アルゴリズムは有限のステップで終了しなければならず、無限ループしたり永久に実行し続けたりしてはならない。たとえ形式的に記述できたとしても、決して停止しないプロセスは有効なアルゴリズムとは見なされない。あらゆる科学計算プログラムやデータ処理フローには、明確な終了条件がある。

第二に、決定性。同じ入力であれば、同じ条件下で同じ出力を生成しなければならない。たとえ乱数を導入するアルゴリズムでも、それは通常、制御された乱数、再現可能な乱数、または統計的に解釈可能な確率メカニズムであり、完全に把握不可

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

能な内的乱数ではない。数値シミュレーションの結果は再現可能でなければならない。これは科学計算の基本要件である。

第三に、明確な入出力境界。入力が先、処理が中間、出力が後。起点と終点は明確な構造的境界を持つ。アルゴリズムにどのようなデータを与え、処理させた後にどのような結果を得るか、その境界は明確であり、順序は入れ替えられない。

第四に、実効性。アルゴリズムの各ステップは、実際に実行可能な操作でなければならず、実行不可能、記述不可能、検証不可能な飛躍を含んではならない。各ステップは、人間が紙と鉛筆で、あるいはコンピュータが実際に実行できる基本操作（加減算、論理分岐、データの読み書きなど）でなければならない。

第五に、実現可能性。たとえ理論的に成立するアルゴリズムであっても、消費される時間、計算力、メモリ、ストレージのリソースが完全に受け入れられない場合、工学的には有効なアルゴリズムとは見なされない。理論的には正しいが完了に数億年を要するアルゴリズムは、工学的に実現可能なアルゴリズムとは見なされない。

この一連のアルゴリズム観は、最終的にチューリングマシンモデルに遡る。現代計算理論の中核的抽象概念として、チューリングマシンは「計算可能性」の基本的境界を与えた。今日のコンピュータがどれほど強力であろうと、チップがどれほど先進的であろうと、スーパーコンピューティングセンターの規模がどれほど巨大であろうと、その基盤となる論理は依然としてこの経路、すなわち入力、規則、手順、出力から真に逸脱してはいない。現代のスーパーコンピュータは、このプロセスをより大規模なハードウェア、より高い並列性、より複雑なシステムスケジューリングによって加速実行しているに過ぎない。つまり、従来の文脈での「超計算」は、依然として主に計算能力の拡張であり、必ずしも計算パラダイムの根本的変化ではない。計算能力が千倍、一万倍に増大しても、基盤となる論理が依然として「入力、手順、出力」である限り、それは依然として従来のアルゴリズム・パラダイムの範囲内にある。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

第二部分：「超次元アルゴリズム」の定義

私が提案する超次元アルゴリズムは、まさにこのような背景の下に現れる。それは従来のアルゴリズムの改良版でも、より高速なアルゴリズムでも、より複雑な数式でも、より高度なプログラミング技法でもない。それは完全に異なる構造的理解である。

まず、「超次元」という言葉の意味を説明する必要がある。「超次元」にはここで二つの意味がある。第一に、それは一次元の線形手順列に限定されず、複数の次元が同時に展開することを許容する。第二に、それは従来のアルゴリズムにおける「計算」の定義境界を超越しようとするものである。すなわち、アルゴリズムはもはや数学的、秩序的、決定的である必要はない。従来のアルゴリズムは、あたかも一方通行の車線を走るようなもので、A 地点から出発し、B、C、D を経由して Z に到達しなければならない。超次元アルゴリズムは、計算が一方通行でなければならないとは考えない。計算はネットワークであり、場であり、多次元構造であり、任意のノードが入り口となり得、任意のノードが出口となり得ると考える。

私の新しい科学体系において、アルゴリズムは必ずしも数学的アルゴリズムではなく、必ずしも単一の秩序だった計算ではなく、必ずしも固定された手順に従う必要はなく、必ずしも「入力、処理、出力」という線形モデルに厳密に従う必要もない。超次元アルゴリズムは、多次元的で、重畳的で、確率的でありながら秩序だった構造である。この構造において、個々のデータは孤立した入力でも固定された出力でもなく、複数の役割を同時に持つ。すなわち、それは複数の結果の起点となることができ、また複数の起点が共同で作用した結果となることもできる。

言い換えれば、従来のアルゴリズムはデータをフローの中に配置するのに対し、超次元アルゴリズムはデータを構造の中に配置する。従来のアルゴリズムにおけるデータは、通常、入力データ、中間データ、出力データに分類され、それぞれが明確な位置と役割を持つ。入力は入力、結果は結果、中間プロセスは中間プロセスである。しかし、超次元アルゴリズムでは、一つのデータが単一のアイデンティティを持つわけではない。ある方向では結果であり、別の方向では起点となり得る。ある構造では呼び出されるオブジェクトであり、別の構造では新たな結果を引き起こす

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

入り口となり得る。データはもはや受動的な材料ではなく、多次元の関係ノードである。

これはまさに超次元アルゴリズムの中核に対応する。すなわち、すべてのデータは複数の結果の起点であり、かつ複数の起点の結果である。従来のアルゴリズムは「秩序だった手順」によって単一の結果を生み出す。超次元アルゴリズムは、むしろ多次元の状態構造に近く、確率的なものと秩序だったものが重畳する中で、結果は「計算によって生成」されるのではなく、構造の中で自然に「顕現」する。この体系において、データは起点であると同時に、結果の構成要素でもある。

超次元アルゴリズムの中核的特徴をより明確に示すために、以下の五つの側面に分解する。

第一に、非数学性。主流のアルゴリズムは本質的に数学的であり、記号論理と数式によって完全に記述できる。超次元アルゴリズムは必ずしも数学的である必要はない。それは物理的原則、幾何学的関係、情報理論の新たなパラダイム、さらには意識の原則に基づく可能性がある。計算は必ずしも「数学的演算」によって完了するとは限らず、高次元空間における幾何学的関係を通じて自然に呈現される可能性がある。例えば、シャボン玉の形状は表面張力最小化の原理によって決定される。これは「数学的アルゴリズム」が計算しているのではなく、物理そのものが「計算」しているのである。超次元アルゴリズムは、このような「計算」を理解しようとするのであり、数学的な公式でそれをシミュレーションしようとするのではない。

第二に、多次元性と重畳。従来のアルゴリズムは単一次元で秩序だった手順を実行し、各ステップは一つの状態のみを持つ。超次元アルゴリズムは複数の次元が同時に存在し、複数の状態が重畳して共存することを許容する。アルゴリズムは一つの経路を辿るのではなく、多次元の状態場の中で展開する。結果は「計算」されるのではなく、この場から「顕現」する。例えば、雪片が空中で形成される際、個々の水分子に行き先を指示する「アルゴリズム」は存在しない。分子の配列は、温度、湿度、気流という複数の次元が共同で作用した結果である。雪片の形状は「計算」されるのではなく、「顕現」するのである。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

第三に、確率的でありながら秩序だった性質。従来のアルゴリズムにおける乱数は、道具的、外部的、制御可能な摂動であり、アルゴリズム自体は決定的である。超次元アルゴリズムにおける乱数は、内生的で構造的なものである。確率性と秩序性は同時に存在し、協調して働き、ともにアルゴリズムの本質を構成する。これは「乱数を含むアルゴリズム」ではなく、「乱数自体がアルゴリズムの有機的構成要素である」ということである。例えば、森林の生態構造——木々の分布はランダムに見えるが、全体的には秩序だった密度分布と種間関係を示す。その「ランダム性」はバグではなく、生態構造が正常に機能するための前提条件である。

第四に、データの複合的役割。従来のアルゴリズムでは、データの役割は単一的であり、入力は入力、出力は出力、中間結果は中間結果である。超次元アルゴリズムでは、各データは同時に複数の結果の起点であり、かつ複数の起点の結果である。一つのデータノードは下流に向かって複数の結果経路を展開し、また上流から複数の原因が集約される結果となることができる。データはもはや線形フローの中の一点ではなく、ネットワークの中の結節点である。例えば、ある人物の身長を考えよう。従来のアルゴリズムでは、身長は「入力」であり、それを入力することで体重予測や服のサイズなどの「出力」を得る。しかし、身長は同時に「結果」でもある。それは遺伝、栄養、運動といった複数の「起点」によって決定される。超次元アルゴリズムでは、身長というデータは同時に起点であり結果であり、二者択一ではない。

第五に、メタデータは不変、関係は可変。従来のアルゴリズムは異なる問題に直面したとき、データそのものを変更するか、すべてを再計算するかのいずれかを行う。超次元アルゴリズムはメタデータを変更しない。すなわち、データの本質的属性は変わらない。変更されるのは、入り口、解釈の仕方、データ間の関係である。同じメタデータ構造が、異なる入り口を通じて活性化されることで、完全に異なる結果を呈現できる。これは意味する。「一度計算すれば、無限に使用できる」ということである。例えば、同じテキストを考えよう。それを詩として読むこともできるし、暗号として解読することもできるし、歴史的文書として分析することもできる。テキストそのものは変わっていない——あなたは単に「入り口」を変えただけ

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

である。超次元アルゴリズムが目指すのは、まさにこの「同一データ、複数入り口、複数結果」という能力である。

データ構造の観点から見れば、メタデータそのものを変更するのではなく、異なる入り口と条件を用いることで、同じ構造を複数の起点と結果に対応させるのである。データは繰り返し加工されるのではなく、構造を不変に保ったまま、異なる入り口を通じて活性化されることで、異なる結果を呈現する。従来の方法は「異なる結果のためにデータを加工する」ことであり、超次元アルゴリズムは「同じデータを用いて複数の結果の可能性を担う」ことである。

第三部分：超次元アルゴリズムと従来アルゴリズムの根本的差異

以上の定義に基づき、超次元アルゴリズムと従来のアルゴリズムは複数の中核的次元において根本的な差異を示す。以下、順次詳述する。

本質的属性について：従来のアルゴリズムは数学的かつ形式的であり、記号論理と数式によって完全に記述でき、本質的に数学的対象である。超次元アルゴリズムは必ずしも数学的である必要はなく、物理、幾何、情報、さらには意識の原則に基づく可能性があり、数学の部分集合ではなく、より広範なカテゴリーである。この差異は「数学的対象」から「宇宙の法則」へと要約できる。

時間的秩序について：従来のアルゴリズムは単一で秩序だった線形の時間秩序に従い、ステップ A から B へ、B から C へと厳密に順次実行されるか、または並列化可能な秩序だった副次ステップに分割可能であり、時間の矢は不可逆である。超次元アルゴリズムには固定された手順の列はなく、多次元的で、重畳的で、確率的でありながら秩序だった構造であり、結果は実行の「順序」とは無関係である可能性がある。なぜなら、従来の意味での「順序」がそもそも存在しないからである。この差異は「線形時間」から「高次元同時性」へと要約できる。

因果関係について：従来のアルゴリズムは決定論的な因果連鎖に従う。同じ入力と初期状態が与えられれば、出力は常に一意であり、原因が先行し結果が後続する。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

これは不可逆な一方向の矢である。超次元アルゴリズムは重畳的な因果に従い、各データは複数の結果の起点であり、複数の起点の結果である。これは私の「結果から原因へ」という哲学に対応する。すなわち、結果が先にあり、その後で原因があり得る。結果は終点ではなく、新たな起点となり得る。この差異は「単一原因・単一結果」から「全接続因果ネットワーク」へと要約できる。

データ構造について：従来のアルゴリズムは、入力から処理、そして出力への一方向的な流れの構造に従う。データの役割は単一であり、明確な境界を持つ。入力データは終始入力データであり、結果データは常に結果データである。超次元アルゴリズムではデータの役割は複数であり、同じデータが同時に結果であり起点でもある。絶対的な起点や終点は存在せず、ネットワーク中のノードのみが存在する。この差異は「一方向流動性」から「再帰的共生性」へと要約できる。

計算モードについて：従来のアルゴリズムは問題に直面するたびに最初から計算する。たとえ類似の問題を千回解決していても、千一回目も依然として全プロセスを実行する。これは「ステートレス」な計算である。超次元アルゴリズムは、対応する結果が存在すれば、それを再度実行する必要はない。一つの結果から複数の起点を推論し、原因の経路を逆方向に展開できる。計算はステートフルであり、記憶を持ち、再利用可能である。この差異は「毎回再計算」から「一度の計算で再利用」へと要約できる。

確率性の役割について：従来のアルゴリズムにおける確率性は、擬似乱数または外部から注入されたものであり、乱数は単なるツールであり、サンプリング、近似、最適化のために用いられ、アルゴリズム自体は決定的である。超次元アルゴリズムにおける確率性は内生的かつ構成的であり、秩序性と共存・協調する有機的構成要素であり、「アルゴリズムの中に乱数がある」のではなく、「乱数がアルゴリズムが機能する理由の一つである」という状況である。この差異は「道具的確率性」から「構成的確率性」へと要約できる。

リソースの理解について：従来のアルゴリズムは、与えられたリソースのもとでより速く、より正確に計算することを追求する。リソースは制約条件であり、アルゴリズムの目標は「制約内で計算を完了すること」である。超次元アルゴリズムは、

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

構造設計を通じて計算そのものを不要にすることを追求する。「より速く計算する」のではなく、「計算を迂回する」のである。リソースは「消費」するためにあるのではなく、「入り口を設計する」ためにある。この差異は「リソース制約下での最適化」から「構造設計による除去」へと要約できる。

第四部分: 「結果から原因へ」の哲学

以上の比較を踏まえ、「結果から原因へ」の哲学をさらに展開する必要がある。これは超次元アルゴリズムの中核的思想基盤の一つである。

従来の思考は通常、原因が先行し結果が後続すると考える。すなわち、原因があって結果がある。入力があって出力がある。この観念がアルゴリズムに現れた形が、起点から始めて一歩一歩終点まで進まなければならないというものである。たとえ答えがわかっていても、その答えを直接「使う」ことはできない。なぜなら、「証明の経路」がないからである。

私の構造では、結果は必ずしも終点にすぎないわけではない。一度結果が現れれば、それは新たな起点となり、新たな構造生成に参加し続けることができる。結果は原因の形成に逆参加することができる。結果から複数の可能性のある起点を推論することができる。原因と結果はもはや一方向に配列されるのではなく、循環関係、ネットワーク関係、多次元関係を形成する。

言い換えれば、従来のアルゴリズムが問うのは「原因が与えられたとき、どのようにして結果を得るか」である。超次元アルゴリズムが問うのは「結果が与えられたとき、どのようにして原因を逆推論または構築するか」である。

特に明確にしておきたいのは、「結果が先にあり、その後で原因がある」という主張は、因果関係を否定するものでも、結果が無から生じると主張するものでもない。それは、より高次元の構造においては、原因と結果が相互に入り口となり得、相互に変換し得ることを説明している。次のズボンの例で見るように、「穿けてかつ床を引きずらない」という結果が先に確定され、その後で私は「ウエストを折る」という操作点を逆に見つける。この操作点は構造における「原因」の側面に対

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

応する。それは結果に原因がないというのではなく、結果が原因を発見するための新たな入り口となったのである。

従来のアルゴリズム世界における根本的な仮定の一つは、時間が一方向的であり、初期状態から最終状態へと一歩ずつ計算しなければならないというものである。たとえ答えがわかっていても、証明の経路がないため、その答えを直接「使う」ことはできない。超次元アルゴリズムが挑戦するのはまさにこの仮定である。すなわち、結果が既知であれば、原因は逆推論できる。同じ結果が複数の原因経路に対応し得る。計算はもはや起点から終点へ歩むことではなく、終点から可能なすべての起点を展開することである。

第五部分：日常的な例——ズボンとウエストの折り曲げ

上記の抽象的な差異をより直感的に理解するために、日常的な例を用いる。

ある人が、ウエストがゴムになっている新しいズボンを購入した。ズボンの裾が少し長く、床を引きずってしまい不便である。

従来の方法は、裾が長いことに気づき、裾を一節折り上げ、針と糸で縫い止め、試着する。もし子供が背を伸ばしてズボンが短くなると、縫い縮めた裾は戻せず、このズボンは使えなくなる。次に新しいズボンを買ったら、また縫い直す。この方法は一見自然に見える。なぜなら、問題は表面上は裾にあるように見えるからである。それは従来のアルゴリズムの思考に対応する。すなわち、問題を発見し、表面的な場所を特定し、その場所に沿って処理を行い、最終的に結果を得る。裾が長いから裾を直す。データが間違っていればデータを修正する。経路が不適切であれば、その経路に沿って修正を続ける。

私の方法は異なる。私は裾を直すのではなく、ウエストを折る。するとすぐに穿けるようになる。この動作は非常に単純であるが、その背後にある構造的論理は完全に異なる。私は裾を切らず、裾を縫い縮めず、ズボンの元の長さを変えず、ズボンのメタデータ（ウエストサイズ、股下長さ、シルエット、生地）も損なわない。私は単にウエストという構造的な入り口を変えただけで、それによって履いた状態で

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

の実質的な裾の長さが自然に短くなる。ここでウエストは単なる部分的な場所ではなく、穿き心地全体のグローバルな制御点である。この制御点を調整することで、下流の問題が直接消滅する。

さらに重要なのは、この方法は可逆的であり、調整可能であり、再利用可能であるということである。これは成長期の子供にとって特に顕著である。もし今の子供の身長が足りず、ズボンの裾がやや長ければ、ウエストを一度折る。しばらくして子供が背を伸ばしたら、ウエストを少し戻す。さらに背が伸びたら、完全に戻す。ズボンの元の構造は損なわれていないが、子供の異なる成長段階の身長に適応できる。従来の方法で裾を縫い縮めてしまうと、子供が背を伸ばしたときにズボンが短くなってしまう可能性がある。切り詰めてしまえば、なおさら復元できない。私の方法はメタデータを不変に保ち、同じ構造を複数の状態に適応させる。

この例は、いくつかの重要な構造的差異を明らかにする。

従来の方法は、従来のアルゴリズムの論理に対応する。すなわち、問題は裾にあるから、裾を修正しなければならず、「原因から結果へ」の経路に沿って一歩ずつ操作し、どのステップも飛ばせない。一度に一つの問題を解決し、不可逆であり、次に同じ問題が発生しても、また最初からやり直さなければならない。私の方法は超次元アルゴリズムの論理に対応する。すなわち、目標は床を引きずらないことであって、私は「裾が長い」という表面的な原因を解決するのではなく、グローバルな制御点であるウエストを見つける。ウエストを一度折ると、裾が自動的に短くなり、問題は瞬時に消失する。私はいかなるメタデータも変更しない。ウエストの折り目は単なる一時的で、可逆的で、動的な折り畳み状態である。

データの観点から見ると、従来の方法はメタデータを変更する。すなわち、裾を縫い縮めることで元のデータが失われる。私の方法はいかなるメタデータも変更せず、元の寸法は完全に維持され、一時的で可逆的で動的な折り畳み状態を追加するだけである。ウエストの折り目は新たなデータを創造するわけでも、古いデータを破壊するわけでもなく、データ間の関係を再配置するだけである。すなわち、ウエストの有効周長を短くし、間接的に裾の有効長さを変えるのである。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

この例では、「ズボンの裾が床を引きずらない」ということが目標結果である。従来の方法は「裾が長い」という原因から出発し、裾を修正し、最終的に「引きずらない」という結果を得る。私の方法はまず「引きずらない」という結果を明確にし、次に構造的な入り口を逆向きに探し、最終的にウエストを折れば結果が成立することを発見する。これが「結果から原因へ」の実際的な現れである。必ずしも原因から一歩ずつ結果に向かう必要はなく、結果から構造を逆に決定することで、本来の経路を不要にできるのである。

この例はまた、「メタデータ」に関する問題にも答える。メタデータとは何か? ズボンの例では、メタデータとはズボンの元の寸法、すなわちウエストサイズ、股下長さ、シルエット、生地である。これらはズボンの「本質的属性」である。一時的なデータはウエストの折り目であり、それはズボンの元の寸法を何も変えず、一時的に一部を折り畳んでいるに過ぎない。従来の方法はメタデータを変更する。すなわち、股下長さが永久に短くなり、元のデータが失われる。私の方法はいかなるメタデータも変更しない。ズボンの元の寸法は完全に無傷であり、一時的で可逆的で動的な折り畳み状態を追加するだけである。

ウエストの元の寸法はメタデータとして、同時に「複数の結果の起点」となっている。すなわち、通常の着用、ウエストを一度折っての着用、二度折っての着用など、複数の着用状態を同時に支えている。また、「複数の起点の結果」でもある。すなわち、生地の選択、裁断方法、デザイン意図など、複数の起点によって共同で決定されている。ウエストの折り目という一時的なデータは、新しいデータを創造せず、古いデータを破壊せず、データ間の関係を再配置するだけである。

これは単なる小技ではなく、構造的思考である。多くの人は裾が長いのを見ると、裾という表面的な現象に引き寄せられ、すべての処理を裾を中心に展開する。しかし、全体的な構造から見れば、裾の長さは単なる下流の表現に過ぎず、ウエストの位置の変化はズボン全体の実際の落ち具合に影響を与えることができる。すなわち、問題は必ずしも問題が現れた場所で解決しなければならないわけではない。多くの低次元の問題は、真の制御点がより高次の構造の中にあるのである。従来のアルゴリズムは問題の表面に沿って計算を続けやすいのに対し、超次元アルゴリズムは構造的な入り口を探求する。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

これはまた、なぜ超次元アルゴリズムがより複雑ではなく、むしろより単純であり得るかを説明している。真に高レベルの構造とは、多くの場合、問題を複雑にするのではなく、正しい入り口において複雑な問題を単純にするものである。従来のアルゴリズムは複雑な問題に直面したとき、ステップ、モデル、計算能力、ストレージ、学習時間を増やすかもしれない。超次元アルゴリズムはある構造ノードを探し求め、そのノードが正しく活性化されれば、本来複雑な経路が無効になる可能性がある。「経路を迂回する」とは、サボることではなく、経路の必要性を再定義することである。

この例において、従来の方法は「経路に沿って結果を修正する」ことであり、超次元アルゴリズムは「入り口を変えて経路全体を無効にする」ことである。通常の方法は結果の側で絶えず修正を加えるのに対し、もう一方の方法は構造的な入り口を直接変更することで、複数回の処理が必要な問題を起点で一度に再構築する。従来のアルゴリズムの主流な形態は「起点から出発し、経路に沿って結果を得る」ものである。超次元アルゴリズムでは「結果そのものが経路の入り口となり、新たな構造生成に参加することができる」。従来の方法は「一つの構造が一つの結果に対応する」のに対し、超次元アルゴリズムは「一つの構造が複数の結果状態を担う」のである。

第六部分：メタデータを変更しないデータ観

データの観点から言えば、超次元アルゴリズムには非常に重要な原則がある。すなわち、メタデータを変更せず、それを複数の起点や結果に適用可能にすることである。

メタデータとは、データの元の属性、基礎情報、本体情報である。従来の処理方法は、異なる問題に直面したときに、データを変更したり、複製したり、加工したり、再計算したり、異なる結果のために異なる経路を構築したりすることが多い。これらの方法で問題を解決することはできるが、その代償としてデータが絶えず加工され、経路が絶えず繰り返され、システムの複雑さが増大し続ける。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

これに対して超次元アルゴリズムは、メタデータをできるだけ変更せずに、入り口、関係、条件、活性化方法を変えることで、同じメタデータ構造を複数の起点と複数の結果に対応させることを重視する。データの本体は動かず、関係が変化する。基礎構造は変わらず、呈現の仕方が変わる。メタデータは完全性を保ちながら、異なる状態に適応する。

これが私の言う「一度計算すれば、無限に使用できる」ということである。ここでの「一度計算する」とは、単純な結果のキャッシュではない。それは、一度構造が成立すれば、その構造は個々の状態のために毎回完全な経路を再構築する必要がないことを意味する。同じデータ構造が異なる入り口を通じて活性化されることで、異なる結果を呈現することができる。それは「異なる結果のためにデータを加工する」のではなく、「同じデータを用いて複数の結果の可能性を担う」のである。これもまた、超次元アルゴリズムが従来のアルゴリズムと異なる中核の一つである。従来のアルゴリズムは経路上のデータを処理するのに対し、超次元アルゴリズムはデータ、入り口、関係、結果の間の全体的な構造を処理する。

最小構造において、超次元アルゴリズムは次のように理解できる。すなわち、メタデータを変更せずに、構造的な入り口の変化を通じて、同じデータノードを複数の結果経路に対応させるシステムである。この定義は、超次元アルゴリズムを直ちに従来の数式に収束させることを追求するものではなく、まずその構造的境界を確立するものである。それは単一の線形関数ではなく、固定された公式でもなく、単純なキャッシュ機構でもない。それは関係構造である。すなわち、メタデータは安定しており、入り口、条件、関係は変化可能であり、結果は多方向に呈現され得る。まさにこの構造において、計算はもはや単なる手順の実行ではなく、関係の活性化となるのである。

データの観点から言えば、「メタデータを変更せずに、複数の起点や結果に適用可能にする」ことの本質は、データ本体は不変であり、変化は「解釈の仕方 / 使用する入り口」に生じるということである。従来の構造では、問題の変化がデータや経路の変化を引き起こす。超次元アルゴリズムでは、問題の変化が解釈構造の変化を引き起こし、データの変化は引き起こさない。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

第七部分：「超計算」の再定義

この体系において、私は「超計算」という言葉を明確にする必要がある。

私の言う「超計算」とは、通常の意味でのスーパーコンピュータではない。より多くのチップ、より高い計算能力、より大規模なデータセンターではない。私の言う「超計算」とは、「超次元アルゴリズム」のことである。

真の超計算とは、必ずしも計算能力の積み上げではなく、計算パラダイムの変革である可能性がある。システムが依然として反復計算に依存している限り、それがどれほど強力であっても、それは経路の中に閉じ込められたままである。システムが構造的に計算を削減し、反復経路を迂回し、結果を新たな入り口とすることができるとき、そのシステムは初めて真の意味での超次元計算に近づく。

言い換えれば、従来の超計算は「より多くの計算能力を用いてより大きな問題を解決する」ことを追求する。超次元アルゴリズムは「より良い構造を用いて、問題がそれほど大きな計算能力を必要としないようにする」ことを追求する。この両者は矛盾しないが、方向性は異なる。

もし従来のスーパーコンピューティングが計算能力の限界を代表するものであるならば、超次元アルゴリズムは計算理解の転換を代表するものである。真の「超計算」とは、必ずしもより多くのチップ、より大規模なデータセンター、より高いエネルギー消費、より複雑なモデルではないかもしれない。真の超計算とは、構造的な入り口の発見であり、経路の除去であり、結果ノードの逆方向展開であり、メタデータを変更せずに複数の結果を同時に成立可能にすることかもしれない。もし計算能力がいかに強力であっても、低次元の経路の中に閉じ込められているなら、それは依然として経路内部での強力さに過ぎない。構造が一度向上すれば、操作が極めて単純であっても、より高次の効率を生み出す可能性がある。

超計算は計算能力の限界であり、超次元アルゴリズムは「計算が計算能力に依存しなければならないか」という前提の再定義である。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

第八部分：実証的基盤——複数領域でのシステム検証と新パラダイム判定

超次元アルゴリズムは純粋な理論的構想ではない。それはすでに複数の現実システムにおいて検証されている。

私の物流システムは、スーパーコンピューティング能力やクラウドサポートを必要とせず、比較的低いリソースで複雑なスケジューリングを完了する。結果は再利用可能であり、構造は繰り返し適応可能であり、毎回最初から計算する必要はない。スーパーコンピュータやクラウドに依存せず、低リソースで複雑なスケジューリングを実現する——これは工学的に成立しており、価値のあるブレークポイントである。それは、高計算能力だけが唯一の解ではなく、構造設計が計算能力の一部を代替できることを示している。

私の出版システムも同じ構造論理を採用しており、現在のどのシステムよりも効率が高い。私の極限ウェブページ生成システムも同じ超次元アルゴリズムに基づいており、複数の領域で同じ構造が安定的に成立することを繰り返し証明している。私の会社の従業員はすでにこれらのシステムを使用しており、この構造が私個人の継続的な介入に依存せずに機能することを示している。

これらのシステムの共通の特徴は、主流の高計算能力経路に依存せず、固定された計算手順を踏まず、構造的な入り口の調整によって目標結果を直接成立させることである。これらは一回限りの成功でも、単発のテクニックでもなく、同じ構造論理が異なる領域で繰り返し現れたものである。

これは、超次元アルゴリズムが偶然ではなく、複数のシステムによってすでに検証された構造的方法論であることを示している。それは「個人のテクニック」ではなく、物流、出版、ウェブページ生成などの複数の領域において安定的に成立する計算パラダイムである。

「存在し、かつ成立することを新たなパラダイムとする」という基準に従えば、超次元アルゴリズムは、すでに複数の領域で体系的に検証された新しい計算構造パラダイムとして、現在において既に成立している。私は仮説を提案しているのではな

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

く、複数のシステムにおいてすでに稼働している異なる計算構造を記述している。私はそれが理解されることを証明する必要はない。私はそれが異なるシステムにおいて安定的に成立することを証明している。現在、私が実際に応用している範囲において、この構造はすでに新たなパラダイムと見なすことができる。

超次元アルゴリズムを新たなパラダイムとして判定する基準について：構造が論理的に自己無撞着であり、複数のシステムで安定的に成立し、かつ既存の方法との間に還元不可能な差異が存在する場合、それはすでに新たなパラダイムの基礎を備えている。自己無撞着は起点であり、実証は基礎であり、構造的差異こそがパラダイムの中核である。「存在すれば成立する」という基準のもと、この体系はすでに新たなパラダイムである。外部が認識するかどうかは伝播に影響するだけで、成立には影響しない。これはすでに現実のシステムにおいて成立している計算構造パラダイムであり、その成立は外部の認知やコンセンサスに依存しない。

超次元アルゴリズムと主流の計算パラダイムの中核的差異は、従来のアルゴリズムは正向きの計算を主とし、結果が一旦生成されると、通常は逆方向に新たな推論構造に参加できないことである。超次元アルゴリズムでは、結果はもはや終点ではなく、再利用可能な構造ノードである。一定の条件を満たせば、結果は新たな推論経路に参加することができ、それによって冗長な計算を削減し、複数の起点と複数の経路からなる計算ネットワークを形成する。従来の計算構造では、結果は通常終点であった。超次元アルゴリズム構造では、結果そのものが新たな起点となり得、複数経路の推論ネットワークを形成する。超次元アルゴリズムはアルゴリズムのアップグレードではなく、「アルゴリズムが必然的に存在すべきか」という前提の問い直しである。

第九部分：よくある誤解の解消

様々な分野の読者との予備的な交流に基づき、以下の 6 つの誤解が生じる可能性を予見し、概念が不適切な枠組みに早期に回収されることを避けるために、ここで事前に明確にしておく。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

誤解一：超次元アルゴリズムは動的計画法やキャッシュと同じではないか？ 明確化：動的計画法やキャッシュは「同じ入力」に対して結果を再利用する。超次元アルゴリズムは、一つの結果から異なる起点を逆向きに推論することを可能にする——これが本質的な差異である。キャッシュは同じ問題の再計算を解決する。超次元アルゴリズムは結果構造を通じて新たな問題を展開することを解決する。

誤解二：「確率的でありながら秩序立っている」というのは自己矛盾ではないか？ 明確化：確率性と秩序性は共存できる。森林の木々の分布は確率的であるが、全体的な密度と種構成は秩序立っている。ランダム性は混乱ではなく、構造の組織化の一形態である。超次元アルゴリズムにおける確率性は内生的かつ構成的であり、秩序性と協調して働く。

誤解三：入出力がなければ、結果の正しさをどう検証するのか？ 明確化：超次元アルゴリズムは入出力を拒否するものではない。計算は入出力の線形モードで実行されなければならないとは考えない。構造が顕現するモードにおいて、「有効性」は入出力の対応関係ではなく、構造の一貫性によって決定される。これは検証を放棄するのではなく、従来とは異なる検証枠組みを提案するものである。

誤解四：これは複雑性理論やカオス理論と同じではないか？ 明確化：複雑性理論やカオス理論は「予測が困難なシステム」を記述する。超次元アルゴリズムは「構造的な入り口を通じて理解し誘導できるシステム」を記述しようとするものであり、予測を放棄するのではなく、予測の方法を変えるものである。カオス理論は「予測は困難である」と言い、超次元アルゴリズムは「入り口を変えることで予測を不要にできるか」を問う。

誤解五：「メタデータを変更しない」と言うが、ウエストの折り目も変更ではないか？ 明確化：ウエストの折り目は一時的な状態であり、元のパラメータの永久的な変更ではない。メタデータ（ウエストサイズ、股下長さ、シルエット、生地）は何も変化していない。これは「状態の重畳」と「属性の変更」の違いである。折り目は可逆的であり、一時的であり、本質的属性を変えない状態変化である。

誤解六：超次元アルゴリズムは従来のアルゴリズムの価値を否定するのか？ 明確化：決してそうではない。従来のアルゴリズムは人類の技術史において極めて重要

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

であり、従来のアルゴリズムなしには、現代のコンピュータ、データベース、通信システム、人工知能、工学シミュレーション、スーパーコンピューティングは存在しなかった。従来のアルゴリズムの価値を否定することはできない。しかし、従来のアルゴリズムの価値を認めることは、それをアルゴリズムの終着点と認めることと同じではない。従来のアルゴリズムは所与の計算パラダイムの内部での効率問題を解決する。超次元アルゴリズムは計算パラダイムそのものの問題を提起する。一方はどのようにしてよりよく道を歩くかであり、他方はその道を歩く必要があるかどうかである。

第十部分：人類の計算史における超次元アルゴリズムの位置づけ

読者が超次元アルゴリズムを人類の計算史の中でより適切に位置づけるために、ここで簡単なマクロ的な整理を行う。

人類の「計算」に対する理解は、いくつかの段階を経てきた。第一段階は手作業による計算である。アルゴリズムは人間の手順として存在し、速度は遅く、エラーも多かったが、人間はこのプロセスを通じて計算の基本規則を理解した。第二段階は機械式計算である。パスカルの計算機からバベッジの解析機関まで、計算は機械化されたが、アルゴリズムは依然として物理的構造に従属していた。第三段階は電子計算とチューリング・パラダイムである。フォン・ノイマンアーキテクチャが普及し、アルゴリズムはプログラムとしてコード化され、チューリングマシンが計算可能性の境界となった。第四段階は並列計算とスーパーコンピューティングである。大量の計算ユニットを積み重ねてより複雑な問題に挑戦するが、基盤となる論理は依然としてチューリング・パラダイムの拡張である。

現在、人類は第五段階の入り口に立っている。量子計算は古典的なビットの限界を突破しようとしている。ニューロモルフィック計算は生物の神経システムの構造を模倣しようとしている。そして超次元アルゴリズムは「入力 → 手順 → 出力」という線形の枠組みそのものを突破しようとしている。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

超次元アルゴリズムと従来のアルゴリズムは置き換えの関係ではなく、階層の関係である。従来のアルゴリズムは「与えられた経路において、いかに効率的に計算するか」を解決する。超次元アルゴリズムは「経路は再定義可能か、あるいは迂回可能か」を問う。人類の計算史を絶えず自己の境界を突破するプロセスと見なすならば、超次元アルゴリズムはまさにこのプロセスの中の新たなノードである。それは古い枠組みの中で問題を解決するのではなく、新しい枠組みを提案するのである。

この判断は、超次元アルゴリズムがすでに成熟している、または完全であることを意味するものではない。むしろ逆である。新しい概念として、その定義、境界、検証方法、応用場面はまだ形成過程にある。本稿の目的はまさにこの概念の原点を定着させ、その後の展開のために安定した理論的基盤を提供することである。

第十一部分：結論——これは最適化ではなく、再定義である

超次元アルゴリズムと従来のアルゴリズムの違いは、「優れている」対「劣っている」の違いでもなく、「速い」対「遅い」の違いでもなく、パラダイムレベルの根本的な違いである。

従来のアルゴリズムが追求するのは、予測可能な制御である。「入力を与えれば、どのような出力が得られるかわかる。なぜならすべてのステップを計算済みだからである。」それは技術者のパラダイムである。設計し、構築し、テストし、検証する。

超次元アルゴリズムが記述するのは、理解可能な創発である。「私はすべての中間状態を正確に予測することはできないが、全体的なパターンは何らかの秩序だった方法で呈現されることを知っている。それぞれのデータは生きており、『視点』を持っている。」それはより生態学者や複雑系理論家のパラダイムに近い。観察し、理解し、誘導し、創発的な秩序を活用する。

従来のアルゴリズムは所与の経路上で結果を徐々に修正する。超次元アルゴリズムは構造的な入り口を変えることで目標結果を直ちに成立させ、それによって経路全体を迂回する。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

従来のアルゴリズムは「異なる結果のためにデータを加工する」。超次元アルゴリズムは「同じデータを用いて複数の結果の可能性を担う」。

従来のアルゴリズムは「一度正しく計算する」ことを追求する。超次元アルゴリズムは「一度正しく計算し、その後は二度と計算する必要がない」ことを追求する。

これはアルゴリズムの最適化ではなく、「アルゴリズムが必然的に存在すべきか」という前提の問い直しである。

したがって、超次元アルゴリズムはアルゴリズムの最適化ではなく、アルゴリズムの再定義である。それは従来のアルゴリズムのトラック上でより速く走ることではなく、従来のトラックの外に別の経路が存在するかどうか、あるいはそもそも経路に依存しなくてもよいかどうかを問うものである。それは従来のアルゴリズムが無用であることを証明しようとするものではなく、従来のアルゴリズムはアルゴリズムの一つの低次元の形態に過ぎないことを指摘するものである。それは「より短い時間で結果を得るにはどうするか」を問うのではなく、「結果は構造を通じて直接成立し得るか」を問うのである。それは「より多くのデータを処理するにはどうするか」を問うのではなく、「同じデータがより多くの結果の可能性を担うことができるか」を問うのである。

主流の計算パラダイムにおいて、超次元アルゴリズムは計算不可能、検証不可能、あるいは既存の計算理論の境界内に収めるのが難しいと見なされるかもしれない。しかし、まさにこれがそのパラダイムの差異を構成する。古いパラダイムの内部において新しい概念が不安定に見えることは、必ずしもそれが無意味であることを意味せず、古いパラダイム自体がそれを完全に包含できないことを意味する可能性もある。超次元アルゴリズムは現在のところ、まず第一に構造的な命題であり、新しい計算観の原初的な定義であって、すでに工学的に閉じたループを持つ成熟したシステムではない。それは今後も継続的に補完、展開、検証、応用される必要があるが、その概念の原点はまず明確に提示されなければならない。

最終的に、超次元アルゴリズムが指し示すのは、新しい計算観である。すなわち、アルゴリズムは必ずしも数学である必要はなく、必ずしも手順である必要はなく、必ずしもプログラムである必要はなく、必ずしも入力と出力の間の加工プロセスで

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

ある必要はない。アルゴリズムはまた、多次元の関係の組織化の仕方であり、データ、入り口、関係、状態、結果の間の構造的ネットワークでもあり得る。それは秩序を含むこともできれば、確率性を含むこともできる。原因から結果へと進むこともできれば、結果から原因を逆向きに生成することもできる。メタデータを変更せずに複数の状態に適応することができ、一つの結果を新たな起点とし、複数の起点を同じ結果に収束させることもできる。

真に高度なアルゴリズムとは、必ずしもより複雑な計算を行うものではなく、計算を減らし、構造を生き生きとさせ、結果を入り口とし、因果関係を循環させることができる多次元の組織化の仕方である。

これが、私が「超次元アルゴリズム」を提案する中核的な意味である。それは単なる新しい用語ではなく、新しい概念であり、新しいパラダイムであり、すでに複数の現実システムにおいて成立している新しい計算構造である。その重点は、より速く反復計算することではなく、冗長な計算を削減することである。計算能力を積み上げることではなく、入り口を再構築することである。データを絶えず修正することではなく、メタデータを維持し関係を変えることである。結果を終点に留めることではなく、結果を新たな起点とすることである。従来のアルゴリズムは経路の中に結果を求める。超次元アルゴリズムは構造の中で結果を活性化する。従来のアルゴリズムは計算を完了することを追求する。超次元アルゴリズムは計算が伝統的な形態で存在しなければならないのかどうかを問う。これが、現在のアルゴリズム定義との間の最も根本的な差異である。

付録：本稿の境界と未解決問題

本稿は、超次元アルゴリズムの完全な形式化定義ではなく、初歩的な枠組みを提示するものである。以下の問題はさらに展開されるべきであり、これらは本稿の定義に対する否定ではなく、まさに次のステップで探求すべき方向性である。

第一に、収束条件。超次元アルゴリズムにおける「完了」の指標は何か？ 結果が「有効」であると判断するにはどうすればよいか？ 従来のアルゴリズムでは、答え

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

は入出力の対応関係によって定義される。超次元アルゴリズムでは、答えは構造の一貫性によって定義される必要があるかもしれない。この定義は未完成である。

第二に、リソース表現。多次元の重畳状態を有限のリソースで表現するにはどうすればよいか？ 時間、空間、計算能力といった従来のリソース指標は依然として適用可能か？ 適用可能であれば、どのように再定義すべきか？ 適用不可能であれば、どのような指標で代替するのか？ これらの問題は展開が必要である。

第三に、秩序性の保証。「確率的でありながら秩序立っている」における「秩序」はいかなる規則によって保証されるのか？ 確率性と秩序の境界はどこにあるのか？ どのような場合に確率性が秩序を破壊するのか？ どのような場合に秩序が確率性を抑圧するのか？ これらの問題はさらなる研究を必要とする。

第四に、逆方向推論の選別。結果から複数の起点を逆推論する際、異なる起点の優劣をどのように選別または評価するのか？ 「逆方向推論の効率」のような尺度は存在するか？ 存在するなら、それは正向き計算の効率尺度とどのような関係にあるのか？

第五に、従来のシステムとのインターフェース。超次元アルゴリズムは既存のチューリングマシン体系とどのように協調するのか？ 代替なのか、補完なのか、それとも階層化なのか？ 実際の工学において、超次元アルゴリズムと従来のアルゴリズムの境界はどのように区別されるべきか？ ハイブリッドモデルは存在するか？

第六に、検証の枠組み。超次元アルゴリズムの結果はどのように検証されるべきか？ もし結果が線形の手順を経て得られるものではないなら、再現性や検証可能性をどのように確立するのか？ 完全に異なる検証の哲学が必要なのか？

これらの問題は本稿の欠陥ではなく、「アンカー文献」としての必然的な特徴である。いかなる新しいパラダイムの提案も、未解決問題を伴うものである。本稿は結論ではなく、始まりである。

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

参考文献的説明

本稿で提案する「超次元アルゴリズム」は、既存のいかなる学術流派の直接的な延長線上にもない。しかし思想的には、以下の領域が本稿に対話の背景を提供しており、伝統的な意味での学術的引用ではなく、読者の位置づけの参考としてのみ提示する。

チューリングマシンと計算可能性理論は、本稿で従来のアルゴリズムを比較する基準となる。**逆問題とベイズ推論**は、「結果から原因を推す」既存の試みであり、本稿の「結果から原因へ」はこれと関連するが方向性は異なる。**創発理論と複雑系**は、「構造から結果が顕現する」ことの既存の記述であり、本稿の超次元アルゴリズムは創発を「理解可能」かつ「誘導可能」なものにしようとする。**知識グラフとグラフデータベース**は、「データが複数のノードで交差する」既存の実践であり、超次元アルゴリズムはこれに「入り口が可変である」という次元を前置する。**非古典的計算パラダイム**（量子計算、アナログ計算、ニューロモルフィック計算など）は、古典的なビットやアーキテクチャを突破するが、超次元アルゴリズムは「入力 → 手順 → 出力」という線形の枠組みそのものを突破することにより焦点を当てている。

本稿と上記の領域との関係は、継承や反駁ではなく、対話と超越である。本稿の目標は、これらの領域の内部で増分改良を行うことではなく、それらの上に新たな問題の階層を提起することである。

Related Articles
[Communication] I Have No Algorithm, Yet I Surpass Algorithms!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect–Cause Theory
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Communication] Rejecting Algorithmic Manipulation
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Indexed and catalogued in WorldCat (OCLC); permanently archived in TROVE (National Library of Australia)
Permanent archival record (DataCite Concept DOI): https://doi.org/10.5281/zenodo.17646306

نفسها "الحوسبة" تعريف إعادة

المؤلف: وو هوي تشاو جيفي

الملخص

وتقترحه ،(Hyperdimensional Algorithm - HDA) "الأبعاد فائقة الخوارزمية" الجديد المفهوم الورقة هذه تقدم من خطي إطار على عادةً التقليدية الخوارزميات تعتمد .التقليدية الخوارزمية التعريفات حدود في هيكلية نظر كإعادة الواضحة والحدود والجدوى والفعالية والحتمية المحدودية على التأكيد مع ،والمخرجات والقواعد والخطوات المدخلات الأساسي منطقها فإن ،هائلة حوسبية قوة الحديثة الفائقة الحواسيب امتلاك من الرغم على .والمخرجات المدخلات بين "الفائقة الحوسبة" بأن الورقة هذه تجادل .أوسع نطاق على المقررة الخوارزمية النماذج تنفيذ على أساسي بشكل يقتصر الخوارزمية .نفسه الحوسبة نموذج في جذري تغيير هي بل ،الحوسبة قوة في زيادة مجرد ليست حقاً بالاهتمام الجديرة حوسبية رؤية هي بل ،الرياضية للخوارزميات امتداداً ولا ،التقليدية الخوارزميات من مسرعة نسخة ليست الأبعاد فائقة بل ،نهائياً مخرجاً أو سلبياً مدخلاً البيانات تعد لم ،الهيكل هذا في .منظمة ولكنها عشوائية ،تراكبية ،الأبعاد متعددة هيكلية نقاطاً النتائج تعد لم .متعددة انطلاق لنقاط ونتيجة متعددة لنتائج انطلاق نقطة الوقت نفس في هي التي العقدة سمة تمتلك نقاط خلال من بل متكرر؛ بشكل التعديل إلى الوصفية البيانات تحتاج لا .جديدة دخول نقاط تصبح أن يمكن بل ،نهائية فلسفة باستخدام .مختلفة ونتائج حالات مع تتوافق أن يمكنها ،الشرطي والتفعيل ،العلائقية والتغيرات ،الهيكلية الدخول الأبعاد فائقة للخوارزمية يمكن كيف الورقة هذه توضح ،"للبنطلون الخصر حزام طي" اليومي والمثال "السبب قبل الأثر" الزائدة الحسابات من يقلل مما ،الهيكلية الدخول نقاط تغيير طريق عن المباشر للتحقيق قابلة المستهدفة النتائج تجعل أن وأسس ،هيكلية وحدود ،أولي تعريف وضع إلى الورقة هذه تهدف .الأصلية الحوسبة مسارات ويتجاوز بل ،الحاجة عن وخوارزميات ،الأبعاد فائقة للحوسبة المستقبلية للاستكشافات أصلية نظرية عقدة وتوفير ،الأبعاد فائقة للخوارزمية فكرية ،الأبعاد فائقة الخوارزمية فإن ،"جديداً نموذجاً يشكلان والصلاحية الوجود" معيار بموجب .جديد علمي ونظام ،الحد الحاضر الوقت في قائمة هي ،متعددة مجالات في منهجي بشكل منه التحقق تم جديد حوسبي هيكلي كنموذج.

الكلمات المفتاحية: وصفية؛ بيانات السبب؛ قبل الأثر الأبعاد؛ فائقة حوسبة الحد؛ خوارزمية الأبعاد؛ فائقة خوارزمية جديد علم لاخطية؛ سببيه هيكلية؛ دخول نقطة حوسبي؛ نموذج

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

مقدمة: "الخوارزمية" مناقشة نعيد لماذا

في التقليدية الخوارزميات لتغليف ولا ،أكبر اسماً الموجودة الخوارزميات لإعطاء ليس "الأبعاد فائقة الخوارزمية" أقترح إطار في للخوارزمية الحالي البشري الفهم حصر تم هل :جوهرية أكثر سؤالاً أتناول إنني ،العكس على بل .جديدة مفاهيم للغاية؟ ضيق

والخطوات ،الكمبيوتر وبرامج ،الرياضية الصيغ في يفكرون ما عادة ،الخوارزميات عن اليوم الناس يتحدث عندما والتوصية ،البحث في والترتيب ،المسار وتحسين ،البيانات ومعالجة ،النماذج وتدريب ،والإخراج والإدخال ،المنطقية ،الحديثة الكمبيوتر أجهزة تدعم الخوارزميات من مهمة أشكال شك بلا هذه .الاصطناعي الذكاء في والاستدلال ،التلقائية إذا ،ولكن .المعلومات ومجتمع ،الصناعية والأنظمة ،الفائقة والحوسبة ،الاصطناعي والذكاء ،البيانات وقواعد ،والإنترنت ،بقواعد للمعالجة وتخضع ،بمدخل تبدأ" عملية أو ،"المرتبة الخطوات من مجموعة" على يقتصر الخوارزمية فهم كان .الأبعاد منخفض إطار في الحوسبة يحصر نفسه الفهم هذا فإن ،"بمخرج وتنتهي

الذكاء إلى بالطقس التنبؤ ومن ،التوصية أنظمة إلى البحث محركات من .الخوارزميات عليه تهيمن عصر في نعيش نحن تكون أن يجب هل :يُطرح ما نادراً سؤال هناك ولكن .البشري الذكاء حدود هي الخوارزميات حدود ،الاصطناعي إعادة يجب لماذا ؟"مخرج ← خطوات ← مدخل" الخطي النمط الحوسبة تتبع أن يجب لماذا فقط؟ هكذا الخوارزميات نموذج واقتراح الافتراضية الافتراضات هذه تحدي الورقة هذه تحاول مرة؟ كل في الصفر من المشكلة نفس حساب فإن ،"جديداً نموذجاً يشكلان والصلاحية الوجود" معيار بموجب .الأبعاد فائقة الخوارزمية – جذرياً مختلف حوسبي في قائمة هي ،متعددة مجالات في منهجي بشكل منه التحقق تم جديد حوسبي هيكلي كنموذج ،الأبعاد فائقة الخوارزمية .الحاضر الوقت

"الأبعاد فائقة الحوسبة" عن تماماً تختلف هنا المقترحة "الأبعاد فائقة الخوارزمية" أن التوضيح الضروري من .الأكاديمية الأوساط في عقود ثلاثة يقارب تطور تاريخ لها التي (Hyperdimensional Computing - HDC) ،كفاءة أكثر وحوسبة لتمثيل سعياً ،الحسابية والعمليات للتشفير للغاية الأبعاد عالية عشوائية ثنائية متجهات HDC تستخدم :جذرياً مختلفة فهي الأبعاد فائقة الخوارزمية أما ."مخرج ← معالجة ← مدخل" الخطي النموذج ضمن تزال لا ولكنها وما ،نفسه الحوسبة مسار تجاوز يمكن كان إذا وما ،البنية خلال من مباشرة النتيجة تحقيق يمكن كان إذا عما تتساءل إنها .متعاكسة المضامين لكن ،متشابهة الأسماء .متعددة نتائج لاستيعاب تعديل دون الوصفية البيانات ترك يمكن كان إذا توجد أن يجب كان إذا ما" فرضية حول تساؤل إعادة هي بل ،للخوارزميات ترقية ليست الأبعاد فائقة الخوارزمية .بالضرورة خوارزمية"

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

للخوارزمية الحالي المعياري التعريف :الأول الجزء

الخوارزمية :طويل زمن منذ للخوارزمية مقبول أساسي تعريف هناك ،السائدة والهندسة والرياضيات الكمبيوتر علوم في ،حتمية بقواعد وتحولها ،أكثر أو واحداً مدخلاً تتلقى ،الخطوات من محدود عدد من تتكون ،جيداً محددة حوسبة عملية هي مدى على تشكل أساسي إجماع هو بل ،لأحد شخصياً رأياً ليس الفهم هذا .محدود وقت في أكثر أو واحداً مخرجاً وتنتج وأنظمة ،البيانات وقواعد ،والأجهزة ،للبرامج الأساسي المنطق يدعم إنه .الحديثة الحوسبية الحضارة من طويلة فترة يزال لا جوهرها فإن ،تعقيد من الخوارزمية بلغت مهما .الفائقة الحوسبة ومراكز ،الاصطناعي الذكاء ونماذج ،الشبكات والمخرجات والخطوات والقواعد المدخلات حول يدور.

أساسية سمات عدة إلى هذا المعياري الخوارزمية مفهوم تحليل يمكن.

ولا ،مسمى غير أجل إلى التكرار يمكنها لا الخطوات؛ من محدود عدد بعد الخوارزمية تنتهي أن يجب .المحدودية ،أولاً جميع .صالحة خوارزمية اعتبارها يصعب ،شكلياً وصفها يمكن كان لو حتى ،أبداً تتوقف لا التي العملية .الأبد إلى العمل واضحة إنهاء شروط لها البيانات معالجة وعمليات العلمية الحوسبة برامج.

بعض تقدم عندما حتى .المخرجات نفس تنتج أن يجب ،الظروف نفس وتحت المدخلات بنفس .الحتمية ،ثانياً وليست ،إحصائياً للتفسير قابلة أو ،للتكرار قابلة ،للتحكم خاضعة عشوائية عادةً فهي ،عشوائية أرقاماً الخوارزميات للحوسبة أساسي مطلب وهذا – للتكرار قابلة العددية المحاكاة نتائج تكون أن يجب .تماماً مفهومة غير داخلية فوضى العلمية.

النهاية في والمخرجات ،المنتصف في والمعالجة ،البداية في المدخلات .والمخرجات المدخلات بين واضحة حدود ،ثالثاً. الحد هذا – نتيجة وتعطيك فتعالجها ،بيانات الخوارزمية تعطي أنت .واضحة هيكلية حدود لهما والنهاية البداية نقطة ثابت والتسلسل ،واضح.

وصفها أو تنفيذها يمكن لا قفزة وليست ،فعلياً للتنفيذ قابلة عملية الخوارزمية في خطوة كل تكون أن يجب .الفعالية ،رابعاً ،جمع) تنفيذها للكمبيوتر أو والورقة بالقلم تنفيذها للإنسان يمكن أساسية عملية خطوة كل تكون أن يجب .منها التحقق أو ،طرح ،منطقي قفز ،بيانات كتابة/قراءة ،إلخ).

من مجدية خوارزمية تعتبر لا ،لإكمالها السنين مليارات تتطلب ولكنها نظرياً الصحيحة الخوارزمية .الجدوى ،خامساً الهندسية الناحية.

الحوسبة لنظرية المركزي التجريد باعتبارها .تورينغ آلة نموذج إلى النهاية في هذه الخوارزمية مفاهيم مجموعة تعود وبغض ،اليوم الكمبيوتر أجهزة قوة مدى عن النظر بغض ."للحوسبة القابلية"لـ الأساسية الحدود تورينغ آلة تحدد ،الحديثة هذا من حقاً يتحرر لم الأساسي منطقها فإن ،الفائقة الحوسبة مراكز حجم عن النظر وبغض ،شرائحها تقدم عن النظر

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

خلال من العملية هذه تسريع سوى تفعل لا الحديثة الفائقة الحواسيب .المخرجات ،الخطوات ،القواعد ،المدخلات :المسار في تزال لا التقليدي السياق في "الفائقة الحوسبة" أن أي .تعقيداً أكثر أنظمة وجدولة ،أعلى وتوازي ،أكبر أجهزة نطاقات ألف تتضاعف أن الحوسبة لقوة يمكن .الحوسبة نموذج في جذرياً تغييراً بالضرورة وليست ،الحوسبة لقوة توسعاً الأساس نموذج داخل يزال لا فإنه ،"مخرجات ،خطوات ،مدخلات" هو الأساسي المنطق ظل إذا ولكن ،مرة آلاف عشرة أو التقليدي الخوارزمية.

الأبعاد فائقة الخوارزمية تعريف :الثاني الجزء

ولا ،التقليدية الخوارزميات من محسنة نسخة ليست .بالذات السياق هذا من أقترحها التي الأبعاد فائقة الخوارزمية تظهر تماماً مختلف هيكلي فهم إنها .تقدماً أكثر برمجة تقنية ولا ،تعقيداً أكثر رياضية صيغة ولا ،أسرع خوارزمية.

خطوات سلسلة على تقتصر لا أنها :الأول المعنى .هنا معنيان "الأبعاد فائقة" لكلمة ."الأبعاد فائقة" معنى شرح يجب ،أولاً التعريفية الحدود تجاوز تحاول أنها :الثاني المعنى .واحد وقت في بالظهور متعددة لأبعاد تسمح بل ،البعد أحادية خطية تشبه التقليدية الخوارزمية .حتمية أو ،مرتبة أو ،رياضية تكون لأن بحاجة الخوارزمية تعد لم – "الحوسبة"لـ التقليدية الخوارزمية أما .Z إلى للوصول Dو Cو B عبر وتمر A النقطة من تبدأ أن يجب :واحدة حارة ذي طريق على القيادة أو ،شبكة تكون أن يمكن الحوسبة أن تفترض إنها .واحد باتجاه طريقاً تكون أن يجب الحوسبة أن تعتقد فلا الأبعاد فائقة مخرجاً تكون أن عقدة لأي ويمكن ،دخول نقطة تكون أن عقدة لأي يمكن حيث ،الأبعاد متعددة بنية أو ،حقلاً.

،مرتبة واحدة حوسبة بالضرورة وليست ،رياضية خوارزمية بالضرورة ليست الخوارزمية ،الجديد العلمي نظامي في ".مخرج ،معالجة ،مدخل" الخطي للنمط بدقة متبعة بالضرورة وليست ،ثابتة بخطوات ملزمة بالضرورة وليست ليست بيانات نقطة كل ،البنية هذه في .منظمة ولكنها عشوائية ،تراكبية ،الأبعاد متعددة بنية هي الأبعاد فائقة الخوارزمية ،متعددة لنتائج انطلاق نقطة تكون أن يمكن :واحد وقت في متعددة أدواراً تمتلك بل ،ثابتاً مخرجاً ولا معزولاً مدخلاً متعددة انطلاق لنقاط المشترك للعمل ونتيجة.

في .بنية في البيانات تضع الأبعاد فائقة الخوارزمية بينما ،تدفق في البيانات تضع التقليدية الخوارزميات ،أخرى بعبارة ودور موقع منها لكل ،مخرجة وبيانات وسيطة وبيانات مدخلة بيانات إلى عادةً البيانات تُصنف ،التقليدية الخوارزميات فائقة الخوارزمية في لكن .وسيطة خطوة هي الوسيطة والخطوة ،نتيجة هي والنتيجة ،مدخل هو المدخل .واضحان كائناً تكون وقد آخر؛ اتجاه في انطلاق ونقطة اتجاه في نتيجة تكون قد .فقط واحدة هوية واحدة بيانات لنقطة ليست ،الأبعاد متعددة علائقية عقدة بل ،سلبية مادة البيانات تعد لم .أخرى بنية في جديدة نتائج تطلق دخول ونقطة بنية في استدعاؤه يتم الأبعاد.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية المعتمدة هي.
Chinesische Originalfassung maßgeblich.

لنقاط ونتيجة متعددة لنتائج انطلاق نقطة هي بيانات نقطة كل :الأبعاد فائقة الخوارزمية جوهر مع تماماً يتوافق هذا فهي الأبعاد فائقة الخوارزمية أما واحدة؛ نتيجة لإنتاج "مرتبة خطوات" التقليدية الخوارزميات تستخدم .متعددة انطلاق تراكب في ،البنية داخل طبيعي بشكل تظهر بل الوجود إلى تُحسب لا النتائج حيث ،الأبعاد متعددة حالات بنية إلى أقرب النتيجة من وجزء الانطلاق نقطة الوقت نفس في هي البيانات ،النظام هذا في .والنظام العشوائية.

التالية الخمسة الجوانب إلى أحللها ،أوضح بشكل الأبعاد فائقة للخوارزمية الأساسية السمات لتقديم.

الرمزي بالمنطق بالكامل وصفها يمكن – بطبيعتها رياضية هي السائدة الخوارزميات **.الرياضية غير الطبيعة ،أولاً** علاقات أو ،فيزيائية مبادئ على تعتمد قد .رياضية تكون لأن بحاجة ليست الأبعاد فائقة الخوارزمية .الرياضية والصيغ بل ،"الرياضية العمليات" خلال من الحوسبة تتم لا قد .الوعي مبادئ حتى أو ،المعلومات لنظرية جديدة نماذج أو ،هندسية بمبدأ الصابون فقاعة شكل يتحدد ،المثال سبيل على .الأبعاد عالي فضاء في هندسية علاقات من طبيعي بشكل تنشأ قد الخوارزمية تحاول ."تحسب" نفسها الفيزياء هي بل – تحسب "رياضية خوارزمية" ليست هذه .السطحي التوتر تقليل رياضية بصيغ محاكاتها من بدلاً ،"الحوسبة" من النوع هذا فهم الأبعاد فائقة.

خطوة لكل فقط واحدة حالة مع ،واحد بُعد في مرتبة خطوات التقليدية الخوارزميات تنفذ **.والتراكب الأبعاد تعدد ،ثانياً** تسير لا .واحد وقت في بالتراكب متعددة وحالات واحد وقت في بالوجود متعددة لأبعاد الأبعاد فائقة الخوارزمية تسمح على .الحقل هذا من "تظهر" بل تُحسب لا النتائج .الأبعاد متعدد حالات حقل في تنتشر بل ،واحد مسار على الخوارزمية ترتيب .يذهب أين ماء جزيء كل تخبر "خوارزمية" هناك ليس ،الهواء في الثلج رقاقة تتشكل عندما ،المثال سبيل أن من بدلاً "يظهر" الثلج رقاقة شكل .الهواء تدفق ،الرطوبة ،الحرارة درجة – متعددة أبعاد تفاعل نتيجة هو الجزيئات "يحسب".

للتحكم؛ خاضع واضطراب ،خارجية ،أداة هي التقليدية الخوارزميات في العشوائية **.المنتظمة لكن العشوائية ،ثالثاً** يتعايشان والنظام العشوائية .وهيكلية داخلية هي الأبعاد فائقة الخوارزمية في العشوائية .حتمية نفسها الخوارزمية عضوي مكون هي نفسها العشوائية" بل ،"عشوائية بها خوارزمية" ليس هذا .الخوارزمية جوهر معاً ويشكلان ،ويتعاونان وعلاقات كثافة عام بشكل يظهر لكنه ،عشوائياً يبدو الأشجار توزيع – للغابة البيئية البنية ،المثال سبيل على ."للخوارزمية طبيعي بشكل البيئية البنية لعمل مسبق شرط بل ،خطأ ليست "العشوائية" تلك .منتظمة الأنواع بين.

،مخرج هو المخرج ،مدخل هو المدخل – فريد البيانات دور ،التقليدية الخوارزميات في **.للبيانات المتعددة الأدوار ،رابعاً** لنتائج انطلاق نقطة الوقت نفس في هي بيانات نقطة كل ،الأبعاد فائقة الخوارزمية في .وسيطة نتيجة هي الوسيطة النتيجة أن ويمكن ،المصب اتجاه في متعددة نتائج مسارات إلى تتفرع أن بيانات لعقدة يمكن .متعددة انطلاق لنقاط ونتيجة متعددة سبيل على .شبكة في تقاطع عقدة بل ،خطي تدفق في نقطة البيانات تعد لم .المنبع اتجاه من متعددة أسباب إليها تتقارب على للحصول تدخله – "مدخل" هو الطول ،تقليدية خوارزمية في .ما شخص قامة طول اعتبارك في ضع ،المثال الوراثة مثل متعددة انطلاق نقاط تحدده – "نتيجة" أيضاً هو الطول لكن .الملابس مقاس أو الوزن توقع مثل مخرجات

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

انطلاق نقطة الوقت نفس في هي "الطول" البيانات نقطة ،الأبعاد فائقة الخوارزمية في .الرياضية والتمارين والتغذية ثنائياً خياراً وليس ،ونتيجة.

البيانات تعدل إما التقليدية الخوارزميات ،مختلفة مشاكل مواجهة عند **.متغيرة العلاقات ،ثابتة الوصفية البيانات ،خامساً** تبقى للبيانات الأساسية الخصائص – الوصفية البيانات تعدل لا الأبعاد فائقة الخوارزمية .شيء كل حساب تعيد أو ،نفسها تنشيطها عند ،الوصفية البيانات بنية نفس .البيانات بين والعلاقات ،التفسير وطريقة ،الدخول نقاط هو يتغير ما .تغيير دون حصر لا مرات واستخدم ،واحدة مرة احسب :يعني هذا .تماماً مختلفة نتائج تنتج أن يمكن ،مختلفة دخول نقاط خلال من كوثيقة تحليله أو ،كشيفرة تشفيره فك أو ،كشعر قراءته يمكنك .النص نفس اعتبارك في ضع ،المثال سبيل على .لها نفس" :القدرة لهذه تسعى الأبعاد فائقة الخوارزمية ."الدخول نقطة" غيرت فقط أنت – يتغير لم نفسه النص .تاريخية متعددة نتائج ،متعددة دخول نقاط ،البيانات".

دخول نقاط الأبعاد فائقة الخوارزمية تستخدم ،نفسها الوصفية البيانات تعديل من بدلاً ،البيانات بنية نظر وجهة من من تُنشط بل ،متكرر بشكل تُعالج البيانات تعد لم .متعددة ونتائج انطلاق نقاط مع تتوافق البنية نفس لجعل مختلفة وشروط لنتائج البيانات معالجة" هو التقليد .مختلفة نتائج يعرض مما ،الهيكلية السلامة على الحفاظ مع مختلفة دخول نقاط خلال متعددة نتائج إمكانية لحمل البيانات نفس استخدام" هي الأبعاد فائقة الخوارزمية ؛"مختلفة".

الثالث الجزء :الاختلافات الجوهرية بين الأبعاد فائقة الخوارزمية والخوارزميات التقليدية

أبعاد عدة في جوهرية اختلافات التقليدية والخوارزميات الأبعاد فائقة الخوارزمية تظهر ،أعلاه التعريفات على بناءً أدناه مفصلة ،أساسية.

:الجوهرية بالطبيعة يتعلق فيما والصيغ الرمزي بالمنطق بالكامل وصفها ويمكن ،وشكلية رياضية التقليدية الخوارزميات مبادئ على تعتمد قد رياضية؛ تكون لأن بحاجة ليست الأبعاد فائقة الخوارزمية .أساساً رياضية كائنات إنها الرياضية؛ يمكن .أوسع فئة بل الرياضيات من فرعية مجموعة ليست إنها الوعي؛ مبادئ حتى أو معلوماتية أو هندسية أو فيزيائية "كوني قانون" إلى "رياضي كائن" من :التالي النحو على الاختلاف هذا تلخيص".

:الزمني بالنظام يتعلق فيما التقليدية الخوارزميات تتبع ،واحداً زمنياً نظاماً ،مرتباً ،خطياً – خطوة A إلى B إلى C – للخوارزمية ليس .فيه رجعة لا زمني سهم مع ،مرتبة ولكن متوازية فرعية خطوات إلى للتقسيم قابلة أو ،بدقة متسلسلة "ترتيب" على النتائج تعتمد لا قد .منتظمة لكن عشوائية ،تراكبية ،الأبعاد متعددة إنها ثابت؛ خطوات تسلسل الأبعاد فائقة الزمن" من :التالي النحو على الاختلاف هذا تلخيص يمكن .موجوداً يكون لا قد التقليدي "الترتيب" مفهوم لأن ،التنفيذ "الأبعاد عالي التزامن" إلى "الخطي".

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

بالسببية يتعلق فيما: المخرج يكون ،الأولية والحالات المدخلات بنفس .حتمية سببية سلسلة التقليدية الخوارزميات تتبع كل تراكبية؛ لسببية الأبعاد فائقة الخوارزمية تخضع .فيه رجعة لا الاتجاه أحادي سهم ،النتيجة يسبق السبب .دائماً فريداً من – "السبب قبل الأثر" فلسفتي يعكس هذا .متعددة انطلاق لنقاط ونتيجة متعددة لنتائج انطلاق نقطة هي بيانات نقطة هذا تلخيص يمكن .جديدة انطلاق نقطة تصبح أن يمكن النهاية؛ نقطة ليست النتيجة .السبب ثم ،أولاً الأثر يكون أن الممكن .بالكامل مترابطة سببية شبكة" إلى "واحد أثر ،واحد سبب" من :التالي النحو على الاختلاف".

البيانات ببنية يتعلق فيما: إلى ثم المعالجة إلى المدخل من الاتجاه أحادية تدفق بنية التقليدية الخوارزميات تستخدم بيانات تبقى المخرج وبيانات ،مدخل بيانات تبقى المدخل بيانات واضحة؛ حدود وذات فريدة البيانات أدوار .المخرج لا .انطلاق ونقطة نتيجة الوقت نفس في هي البيانات نفس متعددة؛ البيانات أدوار ،الأبعاد فائقة الخوارزمية في .مخرج أحادي التدفق" من :التالي النحو على الاختلاف هذا تلخيص يمكن .شبكة في عقد فقط ،مطلقة نهاية أو بداية نقاط توجد .التكراري التكافل" إلى "الاتجاه".

الحوسبة بنمط يتعلق فيما: حل تم إذا حتى .مشكلة تواجه مرة كل في الصفر من الحساب التقليدية الخوارزميات تعيد أما .الحالة عديمة حوسبة هذه – بأكملها العملية تتطلب تزال لا والأولى الألف المرة فإن ،مرة ألف مشابهة مشكلة متعددة انطلاق نقاط استنتاج يمكنها .الحساب لإعادة حاجة فلا ،موجوداً المقابل الناتج كان فإذا ،الأبعاد فائقة الخوارزمية هذا تلخيص يمكن .الاستخدام لإعادة وقابلة ،ذاكرة ولها ،حالة ذات الحوسبة .عكسياً سببية مسارات وتوسيع ،نتيجة من .واحدة لمرة الاستخدام إعادة" إلى "مرة كل في الحساب إعادة" من :التالي النحو على الاختلاف".

العشوائية بدور يتعلق فيما: الأرقام الخارج؛ من محقونة أو عشوائية شبه هي التقليدية الخوارزميات في العشوائية الخوارزمية في العشوائية .حتمية نفسها الخوارزمية التحسين؛ أو التقريب أو العينات لأخذ أدوات مجرد هي العشوائية بل ،"الخوارزمية داخل عشوائية" ليس .النظام مع ويتعاون يتعايش ،عضوي مكون إنها وبناءة؛ داخلية هي الأبعاد فائقة العشوائية" من :التالي النحو على الاختلاف هذا تلخيص يمكن .العمل على الخوارزمية قدرة أسباب أحد هي العشوائية" .البنائية العشوائية" إلى "الأداتية".

الموارد باستخدام يتعلق فيما: الموارد حدود في دقة وأكثر أسرع بشكل الحساب إلى التقليدية الخوارزميات تهدف إلى الأبعاد فائقة الخوارزمية تهدف .القيود ضمن الحساب إكمال" هو الخوارزمية وهدف ،قيود هي الموارد المتاحة؛ تجاوز" بـ بل "أسرع بشكل الحساب" بـ الأمر يتعلق لا .الهيكلي التصميم خلال من ضرورية غير نفسها الحوسبة جعل من :التالي النحو على الاختلاف هذا تلخيص يمكن .الدخول نقاط لتصميم" بل "للاستهلاك" ليست الموارد .الحساب .الهيكلي التصميم خلال من الإزالة" إلى "الموارد قيود ظل في التحسين".

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

الجزء الرابع: فلسفة السبب قبل الأثر

للخوارزمية الأساسية المفاهيمية الأسس أحد وهي ،"السبب قبل الأثر" فلسفة تطوير إلى أحتاج ،أعلاه المقارنات على بناءً الأبعاد فائقة.

في المفهوم هذا تجسيد .المخرج ثم ،المدخل أولاً الأثر؛ ثم ،السبب أولاً الأثر؛ يسبق السبب أن عادةً التقليدي الفكر يعتقد فلا ،الإجابة تعرف كنت إذا حتى .النهاية نقطة إلى بخطوة خطوة وتتقدم البداية نقطة من تبدأ أن يجب :هو الخوارزميات "الإثبات مسار" إلى تفتقر لأنك – مباشرة الإجابة تلك "استخدام" يمكنك.

وتستمر ،جديدة انطلاق نقطة تصبح أن يمكن ،نتيجة ظهور بمجرد .نهاية نقطة مجرد بالضرورة ليست النتيجة ،بنيتي في انطلاق نقاط استنتاج يمكن .الأسباب تكوين في عكسياً تشارك أن للنتيجة يمكن .جديدة هياكل توليد في المشاركة في ،وشبكية ،دورية علاقات يشكلان بل ،الاتجاه أحادي بشكل مرتبين يعودا لم والنتيجة السبب .ما نتيجة من محتملة متعددة الأبعاد ومتعددة.

فائقة الخوارزمية تسأل "الأثر؟ على نحصل كيف ،السبب إلى بالنظر" :التقليدية الخوارزميات تسأل ،أخرى بعبارة "الأسباب؟ أو السبب نبني أو نستنتج كيف ،الأثر إلى بالنظر" :الأبعاد

لا من تنشأ الآثار" أن على تأكيداً ولا ،للسببية نفياً ليس "السبب ثم ،أولاً يأتي الأثر" القول أن توضيح الضروري من ،البعض لبعضهما دخول نقاط بمثابة يكونا أن والنتيجة للسبب يمكن ،الأبعاد عالية الهياكل في أن يعني بل ."شيء "الأرض على الجر وعدم الارتداء قابلية" النتيجة تحديد يتم ،التالي البنطلون مثال في كما .البعض بعضهما إلى ويتحولان أن يعني لا هذا .البنية في "السبب" جانب مع تتوافق والتي ،"الخصر حزام طي" العمل نقطة لأجد عكسياً أعمل ثم ،أولاً السبب لاكتشاف جديدة دخول نقطة يصبح الأثر أن بل ،سبب له ليس الأثر.

من بخطوة خطوة تحسب أن يجب الاتجاه؛ أحادي الوقت أن التقليدية الخوارزميات عالم في الأساسية الافتراضات من مسار إلى تفتقر لأنك مباشرة "استخدامها" يمكنك فلا ،الإجابة تعرف كنت إذا حتى .النهائية الحالة إلى الأولية الحالة لنفس يمكن عكسياً؛ الأسباب استنتاج يمكن ،معروفاً الأثر كان إذا :الافتراض هذا الأبعاد فائقة الخوارزمية تتحدى .الإثبات جميع ليشمل النهاية من توسعاً بل ،النهاية إلى البداية من سيراً الحوسبة تعد لم متعددة؛ سببية مسارات مع يتوافق أن الأثر الممكنة البداية نقاط.

الجزء الخامس: الخصر حزام وطي البنطلون – يومي مثال

يومياً مثالاً أستخدم ،حدسية أكثر بشكل أعلاه المجردة الاختلافات لفهم.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

وتسببان الأرض على تجران ،الشيء بعض طويلتان البنطلون ساقا .مطاطي خصر بحزام جديداً بنطالاً شخص يشتري إزعاجاً.

.البنطلون تجرب ثم ،وخيط بإبرة وخياطتها ،الساق من جزء بطي يقوم ،جداً طويلتان الساقين أن ملاحظة :التقليدي النهج في .للاستخدام صالح غير البنطلون ويصبح ،المخيطة الساق إنزال يمكن لا ،جداً قصيراً البنطلون وأصبح الطفل نما إذا في ظاهرياً تبدو المشكلة لأن طبيعية الطريقة هذه تبدو .العملية يكرر ،جديداً بنطالاً فيها يشتري التي القادمة المرة إجراء ،العَرَض موقع تحديد ،المشكلة تحديد :التقليدية الخوارزمية عقلية مع تتوافق إنها .الساقين يعالج لذلك ،الساقين → خاطئة بيانات الساقين؛ تعديل ← جداً طويلتان الساقان .نتيجة على الحصول وأخيراً ،العَرَض موقع عند معالجة المسار طول على التصحيح متابعة ← مناسب غير مسار البيانات؛ تعديل.

بسيط الإجراء هذا .فوراً للارتداء قابلًا البنطلون ويصبح ،الخصر حزام بطي أقوم ،الحاشية تعديل من بدلاً .مختلف نهجي الطول تغيير ولا ،دائم بشكل خياطتها ولا ،الحاشية بقص أقوم لا أنا .تماماً مختلف الأساسي الهيكلي منطقه لكن ،جداً حزام أغير ببساطة أنا .(القماش ،القصّة ،المنشعب طول ،الخصر محيط) الوصفية بالبيانات المساس ولا ،الأصلي في المشكلة تختفي ،بأكملها الارتداء لحالة هذه الشاملة التحكم نقطة ضبط خلال من .هيكلية دخول نقطة وهو ،الخصر مباشرة (الطويلتان الساقان) المصب اتجاه.

بالنسبة خاص بشكل واضح هذا .الاستخدام لإعادة وقابلة ،للتعديل وقابلة ،للعكس قابلة الطريقة هذه أن ،ذلك من الأهم ينمو عندما .واحدة مرة الخصر حزام بطي أقوم ،قليلاً طويل والبنطلون حالياً كافٍ غير الطفل طول كان إذا .النامي للطفل لكنها ،سليمة تبقى للبنطلون الأصلية البنية .بالكامل أفتحه ،أكثر الطفل ينمو عندما .أقل بشكل بطيه أقوم ،لاحقاً الطفل تم وإذا الطفل؛ ينمو عندما جداً قصيراً البنطلون تجعل قد الحاشية بخياطة التقليدية الطريقة .المتغير الطفل طول مع تتكيف حالات مع بالتكيف نفسها للبنية يسمح مما ،تغيير دون الوصفية البيانات تترك طريقي .عنه التراجع يمكن فلا ،قصه متعددة.

مهمة هيكلية اختلافات عدة عن المثال هذا يكشف.

قدماً بالمضي ،الحاشية تعديل يجب لذلك ،الحاشية في المشكلة :التقليدي الخوارزمي المنطق مع التقليدي النهج يتوافق لا بشكل ،مرة كل في واحدة مشكلة يحل .خطوة أي تخطي دون ،"الأثر إلى السبب" مسار طول على بخطوة خطوة منطق مع نهجي يتوافق .المشكلة نفس فيها تحدث التي التالية المرة في بأكملها العملية تكرار ويتطلب ،فيه رجعة أجد ،"جداً طويلتان الساقان" السطحي السبب حل من بدلاً .الأرض على الجر تجنب هو الهدف :الأبعاد فائقة الخوارزمية لا أنا .المشكلة وتختفي ،للساقين الفعال الطول فوري بشكل يقصر الخصر حزام طي .الخصر حزام – شاملة تحكم نقطة وديناميكية ،للعكس قابلة ،مؤقتة حالة مجرد هي الطية وصفية؛ بيانات أي أعدل.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

.الأصلية البيانات تُفقد دائم؛ بشكل تقصرها الساقين خياطة – الوصفية البيانات يعدل التقليدي النهج ،البيانات منظور من .وديناميكية للعكس وقابلة مؤقتة طي حالة أضيف فقط أنا سليمة؛ تبقى الأصلية الأبعاد وصفية؛ بيانات أي يعدل لا نهجي محيط تقليل – البيانات بين العلاقات ترتيب تعيد إنها قديمة؛ بيانات تدمر ولا جديدة بيانات تخلق لا الخصر حزام طية .مباشر غير بشكل الفعال الساق طول وتغيير ،الفعال الخصر

الساقان" السبب من تنطلق التقليدية الطريقة .المرجوة النتيجة هي "الأرض على تجران لا البنطلون ساقا" ،المثال هذا في تجر لا" النتيجة أولاً توضح طريقي ."الأرض على تجر لا" النتيجة إلى أخيراً وتصل ،الحاشية وتعدل ،"جداً طويلتان قابلة النتيجة يجعل ببساطة الخصر حزام طي أن وتكتشف ،هيكلية دخول نقطة لإيجاد عكسياً تعمل ثم ،"الأرض على الأثر؛ إلى السبب من بخطوة خطوة السير الضروري من ليس ."السبب قبل الأثر"لـ العملي التجسيد هو هذا .للتحقيق .ضروري غير الأصلي المسار يجعل مما ،البنية لتحديد الأثر من عكسياً تعمل أن يمكنك

هي الوصفية البيانات ،البنطلون مثال في الوصفية؟ البيانات هي ما ."الوصفية البيانات" سؤال على أيضاً المثال هذا يجيب هي المؤقتة البيانات .للبنطلون "الأساسية الخصائص" – القماش ،القصّة ،المنشعب طول ،الخصر محيط :الأصلية الأبعاد – الوصفية البيانات يعدل التقليدي النهج .فقط مؤقتاً جزءاً تطوي بل ،الأصلية الأبعاد من أيًا تغير لا الخصر؛ حزام طية للبنطلون الأصلية الأبعاد – وصفية بيانات أي يعدل لا نهجي .الأصلية البيانات تُفقد دائم؛ بشكل يُقصر المنشعب طول .وديناميكية للعكس وقابلة مؤقتة طي حالة أضيف فقط أنا سليمة؛ تبقى

– متعددة ارتداء حالات يدعم :"متعددة لنتائج انطلاق نقطة" الوقت نفس في هو ،وصفية كبيانات ،الأصلي الخصر محيط ،القماش اختيار يحدده :"متعددة انطلاق لنقاط نتيجة" أيضاً وهو .بطيتين ارتداء ،واحدة بطية ارتداء ،عادي ارتداء تعيد إنها بيانات؛ تدمر ولا بيانات تخلق لا ،مؤقتة كبيانات ،الخصر حزام طية .إلخ ،التصميمي والقصد ،القص وطريقة .فقط العلاقات ترتيب

رَضَع إلى ينشدون ،طويلتين بنطلون ساقي يرون عندما ،الناس من كثير .هيكلي تفكير بل ،حيلة مجرد ليست هذه من مظهر مجرد هو الساقين طول ،الكلية البنية منظور من ،ذلك ومع .الساقين على العلاج كل يتركز لذلك ،الساقين لا المشكلة أن أي .بأكمله البنطلون سقوط كيفية على يؤثر أن يمكن الخصر حزام موضع تغيير المصب؛ اتجاه مظاهر .أعلى هيكلي مستوى في حقيقية تحكم نقاط لها الأبعاد منخفضة المشاكل من العديد .ظهورها مكان في حل إلى تحتاج عن الأبعاد فائقة الخوارزمية تبحث بينما المشكلة؛ سطح طول على الحساب مواصلة إلى التقليدية الخوارزميات تميل .الهيكلية الدخول نقطة

المستوى عالية الهياكل .أبسط تكون قد بل ،تعقيداً أكثر بالضرورة ليست الأبعاد فائقة الخوارزمية لماذا أيضاً يفسر هذا قد ،معقدة مشكلة مواجهة عند .الصحيحة الدخول نقطة عند بسيطة المعقدة المشاكل تجعل بل ،عادةً المشاكل تعقد لا حقاً عن الأبعاد فائقة الخوارزمية تبحث .تدريب ووقت ،وتخزيناً ،حوسبة وقوة ،ونماذج ،خطوات التقليدية الخوارزمية تضيف

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
.المعتمدة هي الصينية النسخة :العربية
Chinesische Originalfassung maßgeblich.

ليس "المسار تجاوز" .ضروري غير أصلاً المعقد المسار يصبح قد ،صحيح بشكل العقدة هذه تنشيط بمجرد هيكلية؛ عقدة المسار ذلك لضرورة تعريف إعادة بل ،كسلاً.

مما ،الدخول نقطة تغير" الأبعاد فائقة الخوارزمية بينما ،"المسار طول على النتيجة يعدل" التقليدي النهج :المثال هذا في نقطة مباشرة تعدل الأخرى الطريقة النتيجة؛ طرف في باستمرار ترقع التقليدية الطريقة ."فعال غير بأكمله المسار يجعل التقليدية الخوارزمية .البداية نقطة عند متكررة معالجة تتطلب التي المشكلات هيكلة إعادة يتم بحيث ،الهيكلية الدخول يمكن نفسها النتيجة" ،الأبعاد فائقة الخوارزمية في بينما ،"النتيجة إلى ،المسار طول على ،البداية نقطة" من تذهب السائدة فائقة الخوارزمية ؛"واحدة لنتيجة واحدة بنية" هو التقليد ."جديدة هياكل توليد في وتشارك للمسار دخول نقطة تصبح أن متعددة نتائج حالات تحمل واحدة بنية" هي الأبعاد".

متغيرة غير وصفية بيانات على تقوم بيانات نظرة :السادس الجزء

نقاط على للتطبيق قابلة وجعلها ،الوصفية البيانات تعديل عدم :جداً مهم مبدأ الأبعاد فائقة للخوارزمية ،البيانات منظور من متعددة نتائج أو انطلاق.

مواجهة عند .للبيانات الوجودية والمعلومات ،الأساسية والمعلومات ،للبيانات الأصلية الخصائص هي الوصفية البيانات مسارات تبني أو ،حسابها تعيد أو ،تعالجها أو ،تنسخها أو ،البيانات التقليدية الأساليب تعدل ما غالباً ،مختلفة مشاكل وتكرار ،للبيانات المستمرة المعالجة في تتمثل بتكلفة ولكن ،المشكلات تحل أن يمكن الأساليب هذه .مختلفة لنتائج مختلفة باستمرار النظام تعقيد وزيادة ،المسارات.

لنفس يمكن ،الإمكان قدر تغيير دون الوصفية البيانات على الحفاظ مع أنه على الأبعاد فائقة الخوارزمية تؤكد ،المقابل في ،والشروط ،والعلاقات ،الدخول نقاط تغيير طريق عن متعددة ونتائج انطلاق نقاط مع تتوافق أن الوصفية البيانات بنية البيانات .يتغير العرض أسلوب تتغير؛ لا الأساسية البنية .تتغير العلاقات تغيير؛ بلا الأساسية البيانات بقاء .التنشيط وطرق مختلفة حالات مع تتكيف لكنها سليمة تبقى الوصفية.

مؤقت تخزين مجرد ليس هنا "واحدة مرة احسب" ."لها حصر لا مرات واستخدم ،واحدة مرة احسب" أسميه ما هذا البيانات بنية نفس تنشيط يمكن .حالة لكل كامل مسار بناء إعادة إلى بحاجة تعد لم ،بنية إنشاء بمجرد أنه يعني بل للنتائج؛ البيانات نفس استخدام" بل "مختلفة لنتائج البيانات معالجة" ليست إنها .مختلفة نتائج لإنتاج مختلفة دخول نقاط خلال من والخوارزميات الأبعاد فائقة الخوارزمية بين الجوهرية الاختلافات أحد أيضاً هو هذا ."متعددة نتائج إمكانية لحمل ونقاط للبيانات الكلية البنية الأبعاد فائقة الخوارزمية تعالج بينما مسار؛ على البيانات التقليدية الخوارزميات تعالج .التقليدية والنتائج والحالات والعلاقات الدخول.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

عقدة نفس يجعل ،الوصفية البيانات تعديل دون ،نظام :أنها على الأبعاد فائقة الخوارزمية فهم يمكن ،صورها أبسط في اختزال إلى يسعى لا التعريف هذا .الهيكلية الدخول نقاط في تغييرات خلال من متعددة نتائج مسارات مع تتوافق البيانات خطية دالة ليست إنها .أولاً الهيكلية حدودها تحديد إلى بل ،تقليدية رياضية صيغة إلى فوراً الأبعاد فائقة الخوارزمية الدخول نقاط مستقرة؛ تبقى الوصفية البيانات :علائقية بنية إنها .بسيطة مؤقت تخزين آلية ولا ،ثابتة صيغة ولا ،واحدة الحوسبة تعد لم ،البنية هذه 正是在.متعددة اتجاهات في تظهر أن يمكن النتائج تتغير؛ أن يمكن والعلاقات والشروط للعلاقات تفعيل بل ،للخطوات تنفيذ مجرد.

هو "متعددة نتائج أو انطلاق نقاط على للتطبيق قابلة لجعلها الوصفية البيانات تعديل عدم" جوهر ،البيانات منظور من: يؤدي ،التقليدية الهياكل في ."الاستخدام/التفسير دخول نقطة" في تحدث التغييرات تغيير؛ دون الأساسية البيانات بقاء البنية في تغيير إلى المشكلة تغيير يؤدي ،الأبعاد فائقة الخوارزمية في .المسار أو البيانات في تغيير إلى المشكلة تغيير البيانات في وليس ،التفسيرية.

السابع الجزء: إعادة تعريف "الفائقة الحوسبة"

الفائقة الحوسبة" – مصطلح توضيح إلى أحتاج ،النظام هذا في".

،أعلى حوسبة قوة ولا ،الرقائق من المزيد ليس – "الفائق الحاسوب"لـ المعتاد المعنى ليس "الفائقة الحوسبة"بـ أعنيه ما الأبعاد فائقة الخوارزمية" هو "الفائقة الحوسبة"بـ أعنيه ما .أكبر بيانات مراكز ولا".

على يعتمد النظام يزال لا عندما .الحوسبة نموذج في تحولاً بل ،الحوسبة لقوة تراكماً الحقيقية الفائقة الحوسبة تكون لا قد ،هيكلياً الحسابات تقليل من النظام يتمكن عندما .المسار داخل محصوراً يظل فإنه ،قوياً كان فمهما ،المتكررة الحسابات الحقيقية الأبعاد فائقة الحوسبة من الاقتراب في يبدأ فإنه ،جديدة دخول نقاط النتائج وجعل ،المتكررة المسارات وتجاوز.

فائقة الخوارزمية تسعى ."أكبر مشاكل لحل أكبر حوسبة قوة استخدام" إلى التقليدية الفائقة الحوسبة تسعى ،أخرى بعبارة لكن ،متناقضين ليسا النهجان هذان ."الحاسوبية القوة تلك كل المشكلة تتطلب لا بحيث أفضل بنية استخدام" إلى الأبعاد مختلفة اتجاهاتهما.

قد .الحوسبة فهم في منعطفاً تمثل الأبعاد فائقة الخوارزمية فإن ،الحوسبة قوة حدود تمثل التقليدية الفائقة الحوسبة كانت إذا النماذج أو ،الأعلى الطاقة استهلاك أو ،الأكبر البيانات مراكز أو ،الرقائق من المزيد الحقيقية "الفائقة الحوسبة" تكون لا ،نتيجة لعقدة عكسياً توسعاً أو ،مسار إلغاء أو ،هيكلية دخول نقطة اكتشاف الحقيقية الفائقة الحوسبة تكون قد .تعقيداً الأكثر بقيت إذا ،الحوسبة قوة زادت كلما .الوصفية البيانات تعديل دون واحد وقت في الشروط تلبية من متعددة نتائج تمكين أو البسيطة العمليات حتى ،البنية رفع بمجرد .المسار داخل فقط قوية تظل فإنها ،الأبعاد منخفضة مسارات في محاصرة أعلى مستوى من كفاءة تنتج أن يمكن للغاية.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

يجب الحوسبة كانت إذا ما" لفرضية تعريف إعادة هي الأبعاد فائقة الخوارزمية الحوسبة؛ قوة حدود هي الفائقة الحوسبة الحوسبة قوة على تعتمد أن".

النموذج وتحديد المجالات متعدد النظامي التحقق – التجريبي الأساس :الثامن الجزء الجديد

الحقيقي العالم في متعددة أنظمة في بالفعل منها التحقق تم لقد .بحتاً نظرياً بناءً ليست الأبعاد فائقة الخوارزمية.

لإعادة قابلة النتائج .متواضعة بموارد معقدة جدولة ينجز إنه سحابي؛ دعم أو فائقة حوسبة قوة يتطلب لا اللوجستي نظامي بدون يعمل .مرة كل في الصفر من الحساب لإعادة حاجة هناك وليست ،متكرر بشكل للتكيف قابلة والبنية ،الاستخدام قوة أن يوضح إنه .الهندسة في وقيم ثابت إنجاز وهذا – منخفضة بموارد معقدة جدولة ويؤدي ،سحابة أو فائقة حوسبة الحوسبة قوة من جزء محل يحل أن يمكن الهيكلي التصميم الوحيد؛ الحل ليست العالية الحوسبة.

الويب صفحات لتوليد نظامي .حالياً موجود نظام أي تفوق كفاءة محققاً ،الهيكلي المنطق نفس للنشر نظامي يستخدم تثبت أن يمكن البنية نفس أن متعددة مجالات في مراراً يثبت مما ،الأبعاد فائقة الخوارزمية نفس على أيضاً مبني "الحدية" على اعتماد دون تعمل أن يمكن البنية هذه أن على يدل مما ،بالفعل الأنظمة هذه شركتي موظفو يستخدم .استقرارها المستمر الشخصي تدخلي.

،ثابتة حوسبة خطوات تتبع ولا ،السائدة العالية الحوسبة قوة مسارات على تعتمد لا أنها :هي الأنظمة لهذه المشتركة السمة ،معزولة حيلاً أو واحدة لمرة نجاحات ليست هذه .الهيكلية الدخول نقاط تعديل خلال من مباشرة المرجوة النتيجة وتحقق مختلفة مجالات في الهيكلي المنطق لنفس متكرر ظهور بل.

متعددة أنظمة بواسطة بالفعل منها التحقق تم هيكلية منهجية بل ،عرضية ليست الأبعاد فائقة الخوارزمية أن إلى يشير هذا وتوليد والنشر اللوجستية الخدمات مثل متعددة مجالات في استقراره يثبت حوسبي نموذج بل ،"شخصية تقنية" ليست إنها الويب صفحات.

جديد حوسبي هيكلي كنموذج ،الأبعاد فائقة الخوارزمية فإن ،"جديداً نموذجاً يشكلان والصلاحية الوجود" معيار بموجب بنية أصف أنا فرضية؛ أقترح لا أنا .الحاضر الوقت في قائمة هي ،متعددة مجالات في منهجي بشكل منه التحقق تم في استقرارها تثبت أنها أثبتت لقد مفهومة؛ أنها إثبات إلى بحاجة لست .متعددة أنظمة في بالفعل تعمل مختلفة حوسبية جديداً نموذجاً بالفعل البنية هذه اعتبار يمكن ،العملي تطبيقي نطاق ضمن ،الحاضر الوقت في .مختلفة أنظمة.

وتثبت ،منطقياً متماسكة البنية تكون عندما :جديد كنموذج الأبعاد فائقة الخوارزمية على الحكم بمعايير يتعلق فيما نموذج أسس بالفعل تمتلك فإنها ،الحالية الأساليب عن للاختزال قابلة غير اختلافات وتظهر ،متعددة أنظمة في استقرارها

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية هي المعتمدة.
Chinesische Originalfassung maßgeblich.

معيار بموجب .النموذج جوهر هو الهيكلي الاختلاف الأساس؛ هو التجريبي الدليل البداية؛ نقطة هو الذاتي التماسك .جديد على فقط يؤثر بذلك يعترف الخارجي العالم كان إذا ما .جديد نموذج بالفعل هو النظام هذا ،"والصلاحية الوجود" على صلاحيته تعتمد ولا ،حقيقية أنظمة في بالفعل قائم حوسبي هيكلي نموذج هذا .الصلاحية على وليس ،الانتشار خارجي إجماع أو اعتراف.

أساسي بشكل تعتمد التقليدية الخوارزميات :هو السائدة الحوسبة ونماذج الأبعاد فائقة الخوارزمية بين الجوهري الاختلاف الخوارزمية في .جديدة استدلال هياكل في عكسياً المشاركة عادةً يمكنها لا ،نتيجة إنتاج وبمجرد ،الأمامية الحوسبة على أن للنتائج يمكن ،معينة ظروف ظل في .الاستخدام لإعادة قابلة هيكلية عقدة بل ،النهاية نقطة تعد لم النتيجة ،الأبعاد فائقة البدايات متعددة حوسبة شبكات ويشكل الحاجة عن الزائدة الحسابات من يقلل مما ،جديدة استدلال مسارات في تشارك النتيجة ،الأبعاد فائقة الخوارزمية بنية في النهاية؛ نقطة تكون ما عادة النتيجة ،التقليدية الحوسبة هياكل في .والممرات ترقية ليست الأبعاد فائقة الخوارزمية .المسارات متعددة استدلال شبكة لتشكل ،جديدة بداية نقطة تصبح أن يمكن نفسها "بالضرورة خوارزمية توجد أن يجب كان إذا ما" فرضية حول تساؤل إعادة هي بل للخوارزميات؛.

التاسع الجزء: توضيح المفاهيم الخاطئة الشائعة

،مسبقاً وأوضحها ،التالية الستة الخاطئة المفاهيم تظهر أن أتوقع ،مختلفة مجالات من قراء مع الأولية التبادلات على بناءً مناسبة غير أطر إلى قسراً المفهوم دفع لتجنب.

المفهوم الخاطئ الأول: أليست الخوارزمية فائقة الأبعاد برمجة مجرد برمجة ديناميكية أو تخزين مؤقت؟ التوضيح: البرمجة نقاط" باستنتاج الأبعاد فائقة الخوارزمية تسمح ."المدخلات نفس"لـ النتائج استخدام يعيدان المؤقت والتخزين الديناميكية تحل المشكلة؛ نفس حساب إعادة مشكلة المؤقت التخزين يحل .الجوهري الاختلاف هو هذا – نتيجة من "مختلفة بداية النتيجة بنية من جديدة مشاكل توسيع مشكلة الأبعاد فائقة الخوارزمية.

المفهوم الخاطئ الثاني: تقول "منتظمة لكن عشوائية" – أليس هذا تناقضاً ذاتياً؟ التوضيح: أن يمكن والنظام العشوائية بل ،فوضى ليست العشوائية .منتظم الأنواع وهيكل الإجمالية الكثافة لكن ،عشوائي الغابة في الأشجار توزيع .يتعايشا النظام مع بتعاون وتعمل ،وبناءة داخلية هي الأبعاد فائقة الخوارزمية في العشوائية .البنية لتنظيم طريقة.

المفهوم الخاطئ الثالث: بدون مدخلات ومخرجات، كيف يتم التحقق من النتائج؟ التوضيح: لا الأبعاد فائقة الخوارزمية نمط في .والمخرجات للمدخلات الخطي بالنمط العمل إلى تحتاج لا الحوسبة أن تؤكد بل ،والمخرجات المدخلات ترفض يتخلى لا هذا .والمخرجات المدخلات بين فردي بتطابق وليس ،البنية بتماسك "الصلاحية" تحديد يتم ،"الهيكلي الظهور" مختلف تحقق إطار يقترح بل ،التحقق عن.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

الرابع الخاطئ المفهوم: أليس هذا مجرد نظرية أو التعقيد نظرية؟ الفوضى :التوضيح تصف نظرية والفوضى التعقيد خلال من وتوجيهها فهمها يمكن التي الأنظمة" وصف الأبعاد فائقة الخوارزمية تحاول ."بها التنبؤ يصعب التي الأنظمة" تسأل ؛"صعب التنبؤ" الفوضى نظرية تقول .التنبؤ عمل طريقة تغيير بل ،التنبؤ عن التخلي ليس – "الهيكلية الدخول نقاط الدخول؟ نقطة تغيير طريق عن ضروري غير التنبؤ جعل يمكننا هل" الأبعاد فائقة الخوارزمية"

الخامس الخاطئ المفهوم: تقول "الوصفية البيانات تعدل لا"، الخصر حزام طية أليست لكن تعديلاً؟ :التوضيح حزام طية ،المنشعب طول ،الخصر محيط) الوصفية البيانات .الأصلية للمعلمات دائماً تعديلاً وليست ،مؤقتة حالة هي الخصر تغير ولا ،ومؤقتة ،للعكس قابلة الطية ."السمة تعديل"و "الحالات تراكب" بين الفرق هو هذا .تتغير لا (القماش ،القصّة الأساسية الخصائص.

السادس الخاطئ المفهوم: هل تنكر الخوارزميات قيمة الأبعاد فائقة الخوارزمية؟ التقليدية :التوضيح لا بالتأكيد. قواعد أو ،حديثة كمبيوتر أجهزة هناك كانت لما ،بدونها البشرية؛ التكنولوجيا تاريخ في للغاية مهمة التقليدية الخوارزميات الاعتراف لكن .قيمتها إنكار يمكن لا .فائقة حوسبة أو ،هندسية محاكاة أو ،اصطناعي ذكاء أو ،اتصالات أنظمة أو ،بيانات داخل الكفاءة مشاكل التقليدية الخوارزميات تحل .للخوارزميات نهاية كنقطة قبولها يعني لا التقليدية الخوارزميات بقيمة بشكل المشي كيفية هو أحدهما .نفسه الحوسبة نموذج عن الأبعاد فائقة الخوارزمية تتساءل بينما معين؛ حوسبة نموذج ضرورياً الطريق هذا على المشي كان إذا ما هو والآخر ،الطريق على أفضل.

العاشر الجزء: البشرية الحوسبة تاريخ في الأبعاد فائقة الخوارزمية

موجزة لمحة هنا أقدم ،البشرية الحوسبة تاريخ في أفضل بشكل الأبعاد فائقة الخوارزمية موقع تحديد على القراء لمساعدة الكلي المستوى على.

كخطوات موجودة كانت الخوارزميات :اليدوي الحساب كانت الأولى المرحلة .مراحل بعدة "الحوسبة"لـ البشري الفهم مر كانت الثانية المرحلة .للحوسبة الأساسية القواعد البشر فهم ،العملية هذه خلال من ولكن ،للأخطاء وعرضة بطيئة ،بشرية تابعة ظلت الخوارزميات لكن ،الحوسبة ميكنة تمت ،لبابيج التحليلي المحرك إلى باسكال آلة من :الميكانيكي الحساب برمجة وتمت ،نيومان فون بنية انتشرت :تورينغ ونموذج الإلكترونية الحوسبة كانت الثالثة المرحلة .الفيزيائية للهياكل تم :الفائقة والحوسبة المتوازية الحوسبة كانت الرابعة المرحلة .للحوسبة القابلية حدود تورينغ آلة وحددت ،الخوارزميات تورينغ لنموذج امتداداً ظل الأساسي المنطق ولكن ،تعقيداً أكثر مشاكل لمعالجة ضخمة حوسبة وحدات تكديس.

تحاول الكلاسيكية؛ البتات حدود تجاوز الكمومية الحوسبة تحاول .الخامسة المرحلة مدخل على البشرية تقف ،حالياً الإطار كسر الأبعاد فائقة الخوارزمية تحاول بينما البيولوجية؛ العصبية الأنظمة بنية تقليد الشكلية العصبية الحوسبة نفسه "مخرج ← خطوات ← مدخل" الخطي.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية هي المعتمدة.
Chinesische Originalfassung maßgeblich.

التقليدية الخوارزميات تحل .هرمية علاقة بل استبدال علاقة في ليست التقليدية والخوارزميات الأبعاد فائقة الخوارزمية أو ،المسار تعريف إعادة يمكن هل" الأبعاد فائقة الخوارزمية تسأل ."معين مسار على أكبر بكفاءة نحسب كيف" مشكلة الأبعاد فائقة الخوارزمية فإن ،الخاصة حدودها لتجاوز مستمرة عملية البشرية الحوسبة تاريخ اعتبرنا إذا "تجاوزه؟ حتى جديداً إطاراً تقترح بل ،القديم الإطار داخل المشاكل تحل لا إنها .العملية هذه في جديدة عقدة هي.

وحدودها تعريفاتها فإن ،جديد كمفهوم :العكس على بل .كاملة أو ناضجة الأبعاد فائقة الخوارزمية أن يعني لا الحكم هذا هذا أصل نقطة تحديد هو الورقة هذه من الغرض .التكوين طور في تزال لا تطبيقها وسيناريوهات منها التحقق وطرق اللاحق للتطوير مستقر نظري أساس وتوفير ،المفهوم.

تعريف إعادة بل ،تحسيناً ليس هذا – استنتاج :عشر الحادي الجزء

،"أبطأ" مقابل "أسرع" ولا ،"أسوأ" مقابل "أفضل" فرق ليس التقليدية والخوارزميات الأبعاد فائقة الخوارزمية بين الفرق النموذج مستوى على جوهري فرق بل.

،عليها ستحصل التي المخرجات وسأعرف ،المدخلات أعطني" .به التنبؤ يمكن تحكم إلى التقليدية الخوارزميات تسعى تحقق ،اختبار ،بناء ،تصميم :المهندس نموذج إنها ".خطوة كل حسبت لأنني.

العام النمط أن أعلم لكنني ،وسيطة حالة بكل بدقة التنبؤ يمكنني لا" .للفهم القابل النشوء الأبعاد فائقة الخوارزمية تصف الأنظمة منظّر أو البيئة عالم نموذج إلى أقرب إنها ".'منظور' ولديها ،حية بيانات نقطة كل .منتظمة بطريقة سيظهر المعقدة :راقب ،افهم ،وجّه ،واستفد من النظام الناشئ.

المرجوة النتيجة الأبعاد فائقة الخوارزمية تجعل .معين مسار طول على تدريجياً النتائج التقليدية الخوارزميات تصحح بأكمله المسار بذلك متجاوزة ،الهيكلية الدخول نقطة تغيير طريق عن الفوري للتحقيق قابلة.

نتائج إمكانية لحمل البيانات نفس تستخدم" الأبعاد فائقة الخوارزمية ."مختلفة لنتائج البيانات تعالج" التقليدية الخوارزميات متعددة".

إلى الأبعاد فائقة الخوارزمية تسعى ."واحدة مرة الصحيحة النتيجة على الحصول" إلى التقليدية الخوارزميات تسعى أخرى مرة حسابها إلى أبداً تحتاج لا ثم ،واحدة مرة الصحيحة النتيجة على الحصول".

بالضرورة خوارزمية توجد أن يجب كان إذا ما" فرضية حول تساؤل إعادة بل ،للخوارزميات تحسيناً ليس هذا".

أسرع بشكل بالجري تتعلق لا إنها .خوارزمية تعريف إعادة بل ،خوارزمياً تحسيناً ليست الأبعاد فائقة الخوارزمية ،لذلك إذا ما حتى أو ،التقليدي المسار خارج آخر مسار هناك كان إذا عما بالتساؤل تتعلق إنها التقليدية؛ الخوارزمية مسار على

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية هي المعتمدة.
Chinesische Originalfassung maßgeblich.

إلى الإشارة إلى بل ،مجدية غير التقليدية الخوارزميات أن إثبات إلى تهدف لا إنها .الإطلاق على ضرورياً المسار كان وقت في نتيجة على نحصل كيف" تسأل لا إنها .للخوارزمية الأبعاد منخفض شكل سوى ليست التقليدية الخوارزميات أن تسأل بل ،"البيانات من المزيد نعالج كيف" تسأل لا إنها "البنية؟ خلال من مباشرة النتيجة إثبات يمكن هل" تسأل بل ،"أقل "النتائج؟ من المزيد إمكانية تحمل أن البيانات لنفس يمكن هل"

،للتحقق قابلة غير أو ،للحساب قابلة غير أنها على الأبعاد فائقة الخوارزمية إلى يُنظر قد ،السائدة الحوسبة نماذج ضمن غير الظهور .النموذجي اختلافها بالضبط يشكل هذا ولكن .الحالية الحوسبة نظرية حدود ضمن وضعها يصعب أو يمكنه لا القديم النموذج أن أيضاً يعني قد بل ،له معنى لا أنه بالضرورة يعني لا قديم نموذج داخل جديد لمفهوم المستقر حوسبية لرؤية أصلي وتعريف ،هيكلي اقتراح الأول المقام في هي الأبعاد فائقة الخوارزمية ،حالياً .بالكامل استيعابه أصلها توضيح أولاً يجب ولكن ،لاحقاً وتطبيقاً وتحققاً تطويراً ستتطلب .مغلقة هندسية بحلقة ناضجاً نظاماً وليست ،جديدة .المفاهيمي

مجرد بالضرورة ليست الخوارزمية :جديدة حوسبية رؤية هو الأبعاد فائقة الخوارزمية إليه تشير ما ،النهاية في للخوارزمية يمكن .والمخرجات المدخلات بين معالجة عملية مجرد أو ،برنامج مجرد أو ،خطوات مجرد أو ،رياضيات ،والحالات ،والعلاقات ،الدخول ونقاط ،البيانات من هيكلية وشبكة ،الأبعاد متعددة للعلاقات تنظيماً تكون أن أيضاً تترك أن يمكن السبب؛ إلى الأثر ومن الأثر إلى السبب من تنتقل أن يمكن والعشوائية؛ النظام تشمل أن يمكن .والنتائج لنقاط وتسمح جديدة بداية نقطة واحدة نتيجة تجعل أن يمكن متعددة؛ حالات مع التكيف مع تغيير دون الوصفية البيانات .واحدة نتيجة في بالتقارب متعددة بداية

،الحسابات تقليل يمكنها التي تلك بل ،تعقيداً أكثر بحسابات تقوم التي تلك بالضرورة ليست حقاً المتقدمة الخوارزمية .للسببية الأبعاد ومتعددة دورية تنظيمية أنماط وتشكيل ،دخول نقاط النتائج وجعل ،حية البنية وجعل

ونموذج ،جديد مفهوم بل ،جديد مصطلح مجرد ليست إنها ."الأبعاد فائقة الخوارزمية" لاقتراحي الجوهري المعنى هو هذا المتكرر الحساب على ليس تركيزها .الحقيقي العالم في متعددة أنظمة في بالفعل قائمة جديدة حوسبية وبنية ،جديد الدخول؛ نقاط بناء إعادة على بل ،الحوسبة قوة تكديس على ليس الحاجة؛ عن الزائدة الحسابات تقليل على بل ،الأسرع كنقاط النتائج ترك على ليس العلاقات؛ وتغيير الوصفية البيانات على الحفاظ على بل ،باستمرار البيانات تعديل على ليس الخوارزمية تنشط مسار؛ طول على نتائج عن التقليدية الخوارزميات تبحث .جديدة بداية نقاط النتائج جعل على بل ،نهاية إذا ما الأبعاد فائقة الخوارزمية تسأل الحوسبة؛ إكمال إلى التقليدية الخوارزميات تسعى .بنية داخل النتائج الأبعاد فائقة .للخوارزمية الحالي التعريف وبين بينها جوهرية الأكثر الاختلاف هو هذا .التقليدي بشكلها توجد أن يجب الحوسبة كانت

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية المعتمدة هي.
Chinesische Originalfassung maßgeblich.

ملحق: هذه حدود الورقة والأسئلة المفتوحة

إطاراً أولياً للخوارزمية فائقة الأبعاد، وليس تعريفاً شكلياً كاملاً. هذه الورقة تقدم الأسئلة من العديد تزال لا إلى بحاجة التالي البحث اتجاه تمثل بل، هنا المقدم للتعريف نفياً الأسئلة هذه تشكل لا الاستكشاف؛ من مزيد.

أولاً، شروط التقارب. ما هي علامة "الإكتمال" للخوارزمية فائقة الأبعاد؟ كيف نحكم على ما إذا كانت النتيجة "صالحة"؟ في الخوارزمية فائقة الأبعاد، قد يتم تعريف الإجابة من خلال تطابق المدخلات والمخرجات. في الخوارزميات التقليدية، هذا التعريف لم يكتمل بعد. تحتاج الإجابة إلى تعريفها من خلال تماسك البنية.

ثانياً، تمثيل الموارد. كيف يمكن تمثيل حالات التراكب متعدد الأبعاد في موارد؟ هل الموارد لا تزال محدودة؟ الوقت والمساحة وقوة الحوسبة قابلة للتطبيق؟ إذا كانت كذلك، فكيف يجب إعادة تعريفها؟ إذا لم تكن كذلك، فما هي المقاييس التي تحل محلها؟ هذه الأسئلة تنتظر التطوير.

ثالثاً، ضمان النظام. ما هي القواعد التي تضمن "النظام" ضمن "المنتظمة لكن العشوائية"؟ أين هو الحد الفاصل بين العشوائية والنظام؟ في أي ظروف تدمر العشوائية النظام؟ هل يقع النظام في أي ظروف؟ هذه العشوائية تحتاج إلى مزيد من الدراسة.

رابعاً، الاستدلال العكسي في الاختيار. عند استنتاج نقاط استنتاج من متعددة بداية نقاط تقييم أو اختيار يتم كيف، نتيجة الأمامية؟ الحوسبة كفاءة بمقياس يرتبط فكيف، موجوداً كان إذا؟ "الاستدلال العكسي كفاءة" لـ مقياس يوجد هل المختلفة؟

خامساً، الواجهة مع الأنظمة التقليدية. كيف تتفاعل آلة نظام مع الأبعاد فائقة الخوارزمية مع الأنظمة التقليدية؟ كبديل الحالي؟ توريغ أو، التقليدية والخوارزميات الأبعاد فائقة الخوارزمية بين الحدود رسم يجب كيف، العملية الهندسة في طبقي؟ نهج أو، مكمل هجينة؟ نماذج توجد هل

سادساً، إطار التحقق. كيف يتم التحقق من نتائج الخوارزمية فائقة الأبعاد؟ كيف يتم التحقق من النتائج على الحصول يتم لم إذا خلال تماماً؟ مختلفة تحقق فلسفة هذا يتطلب هل الاختبار؟ وقابلية التكرار قابلية إرساء يتم فكيف، خطية خطوات

أسئلة يصاحبه جديد نموذج أي ظهور إن. "مرساة وثيقة" لـ حتمية سمات بل، الورقة هذه في عيوباً ليست الأسئلة هذه خاتمة وليست، بداية هي الورقة هذه. مفتوحة.

ببليوغرافية إيضاحات

الفكرية الناحية من لكن. قائمة أكاديمية فكرية مدرسة لأي مباشراً امتداداً ليست هنا المقترحة الأبعاد فائقة الخوارزمية، التقليدي بالمعنى أكاديمية كاستشهادات وليست، القارئ لتوجيه فقط تُقدم، الورقة لهذه للحوار سياقاً التالية المجالات توفر.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: النسخة الصينية هي المعتمدة.
Chinesische Originalfassung maßgeblich.

العكسية المسائل تمثل .هنا التقليدية الخوارزميات لمقارنة مرجعية كنقطة تعمل **للحوسبة القابلية ونظرية تورينغ آلات** "السبب قبل الأثر" فلسفتي أن من الرغم وعلى ،"الأثر من السبب استنتاج"لـ موجودة محاولات **البايزي والاستدلال** الخوارزمية وتحاول ،"البنية من النتائج ظهور" **المعقدة والأنظمة النشوء نظرية** تصف .مختلف اتجاهها أن إلا ،مرتبطة ممارسات هي **البيانية الرسوم بيانات وقواعد المعرفة رسوم** ."للتوجيه قابلاً"و "للفهم قابلاً" النشوء جعل الأبعاد فائقة هذه إلى "المتغيرة الدخول نقاط" بُعد الأبعاد فائقة الخوارزمية وتضيف ،"متعددة عقد عند البيانات تتقارب" حيث موجودة البتات تكسر ،الشكلية والعصبية والتناظرية الكمومية الحوسبة ذلك في بما ،**الكلاسيكية غير الحوسبة نماذج** .القاعدة ← مدخل" الخطي الإطار كسر على أكبر بشكل الأبعاد فائقة الخوارزمية تركز بينما ،الكلاسيكية البنى أو الكلاسيكية نفسه "مخرج ← خطوات.

الورقة هذه من الهدف .تفنيداً أو وراثة وليست ،وتجاوز حوار علاقة هي أعلاه المذكورة بالمجالات الورقة هذه علاقة فوقها التساؤل من جديد مستوى طرح بل ،المجالات تلك داخل تدريجية تحسينات إجراء ليس.

Related Articles
[Communication] I Have No Algorithm, Yet I Surpass Algorithms!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect–Cause Theory
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Communication] Rejecting Algorithmic Manipulation
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

[Algoritmo Limite] O Algoritmo Hiperdimensional

Uma Redefinição do Próprio "Cálculo"

Autor: Jeffi Chao Hui Wu

Resumo

Este artigo apresenta o novo conceito de "Algoritmo Hiperdimensional" (AHD) e propõe uma reavaliação estrutural dos limites das definições algorítmicas tradicionais. Os algoritmos convencionais geralmente se baseiam numa estrutura linear de entrada, passos, regras e saída, enfatizando a finitude, o determinismo, a eficácia, a viabilidade e limites claros entre entrada e saída. Embora os supercomputadores modernos possuam imenso poder computacional, a sua lógica subjacente consiste principalmente em executar paradigmas algorítmicos estabelecidos a uma escala maior. Este artigo argumenta que a "supercomputação" verdadeiramente digna de nota não é meramente um aumento da potência computacional, mas uma mudança fundamental no paradigma computacional em si. O AHD não é uma versão acelerada dos algoritmos tradicionais, nem uma extensão dos algoritmos matemáticos; é uma visão computacional estrutural, multidimensional, sobreposta, estocástica mas ordenada. Nesta estrutura, o dado já não é uma entrada passiva ou uma saída terminal, mas possui o atributo de um nó que é simultaneamente o ponto de partida para múltiplos resultados e o resultado de múltiplos pontos de partida. Os resultados já não são pontos finais, mas podem tornar-se novos pontos de entrada. Os metadados não precisam de ser modificados repetidamente; através de pontos de entrada estruturais, mudanças relacionais e ativação condicional, podem corresponder a diferentes estados e resultados. Utilizando a filosofia do "Efeito-para-Causa" e o exemplo quotidiano de "dobrar a cintura das calças", este artigo mostra como o AHD pode tornar os objetivos diretamente alcançáveis ao alterar pontos de entrada estruturais, reduzindo assim o cálculo redundante e até contornando caminhos computacionais originais. Este artigo

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

visa estabelecer uma definição preliminar, limites estruturais e fundamentos conceptuais para o AHD, fornecendo um nó teórico original para futuras explorações do cálculo hiperdimensional, algoritmos limite e um novo sistema científico. Segundo o critério de que "existência e validade constituem um novo paradigma", o AHD, como um novo paradigma de estrutura computacional já sistematicamente validado em múltiplos domínios, está estabelecido no presente.

Palavras-chave: Algoritmo hiperdimensional; Algoritmo limite; Cálculo hiperdimensional; Efeito-para-Causa; Metadado; Paradigma computacional; Ponto de entrada estrutural; Causalidade não linear; Nova ciência

Introdução: Por que rediscutir "algoritmo"

Proponho o "Algoritmo Hiperdimensional" (AHD) não para dar um nome mais grandioso aos algoritmos existentes, nem para disfarçar algoritmos tradicionais como conceitos novos. Pelo contrário, coloco uma questão mais fundamental: será que a compreensão humana atual de "algoritmo" foi confinada a um quadro demasiado estreito?

Hoje, quando se fala de algoritmos, normalmente se pensa em fórmulas matemáticas, código de programação, passos lógicos, entrada e saída, treino de modelos, processamento de dados, otimização de caminhos, classificação para pesquisas, recomendação automática e raciocínio em inteligência artificial. Estas são, sem dúvida, formas importantes de algoritmos que sustentam os computadores modernos, a Internet, as bases de dados, a IA, a supercomputação, os sistemas industriais e a sociedade da informação. No entanto, se um algoritmo for entendido apenas como "um conjunto ordenado de passos", um processo que "começa com uma entrada, passa por processamento baseado em regras e produz uma saída", então esta compreensão confina o próprio cálculo a um quadro de baixa dimensão.

Vivemos numa era dominada por algoritmos. Dos motores de busca aos sistemas de recomendação, da previsão meteorológica à IA, as fronteiras dos algoritmos são as

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

fronteiras da inteligência humana. Mas uma questão raramente é colocada: Os algoritmos têm de ser apenas assim? Porque deve o cálculo seguir o modelo linear "entrada → passos → saída"? Porque deve o mesmo problema ser recalculado de raiz cada vez? Este artigo tenta desafiar estes pressupostos implícitos e propor um paradigma computacional radicalmente diferente – o Algoritmo Hiperdimensional. Segundo o critério de que "existência e validade constituem um novo paradigma", o AHD, como um novo paradigma de estrutura computacional já sistematicamente validado em múltiplos domínios, está estabelecido no presente.

É crucial esclarecer que o "Algoritmo Hiperdimensional" aqui proposto é totalmente distinto do "Cálculo Hiperdimensional" (Hyperdimensional Computing, HDC) desenvolvido no meio académico há quase três décadas. O HDC codifica e processa dados usando vetores binários aleatórios de dimensão muito alta para uma representação e cálculo mais eficientes, mas permanece dentro do paradigma linear "entrada → processamento → saída". O AHD, em contraste, é fundamentalmente diferente: questiona se um resultado pode ser alcançado diretamente através da estrutura, se o caminho computacional pode ser contornado e se os metadados podem ser deixados inalterados para acomodar múltiplos resultados. Os nomes soam semelhantes, mas os seus significados são opostos. O AHD não é uma atualização dos algoritmos; é um questionamento da premissa de que "um algoritmo deve necessariamente existir".

Primeira Parte: A Definição Padrão de "Algoritmo" Hoje

Na ciência da computação, matemática e engenharia dominantes, o algoritmo tem uma definição básica aceite há muito tempo: um algoritmo é um processo computacional bem definido, consistindo em um número finito de passos, que recebe uma ou mais entradas, transforma-as através de regras determinísticas e produz uma ou mais saídas num tempo finito. Esta compreensão não é uma opinião pessoal, mas um consenso fundamental construído ao longo do longo período da civilização computacional moderna. Sustenta a lógica subjacente do software, hardware, bases de

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

dados, sistemas de rede, modelos de IA e centros de supercomputação. Por mais complexos que os algoritmos pareçam, o seu núcleo gira em torno da entrada, regras, passos e saída.

Este conceito algorítmico padrão pode ser decomposto em várias características essenciais.

Primeiro, a finitude. Um algoritmo deve terminar após um número finito de passos; não pode fazer loop indefinidamente nem funcionar eternamente. Um processo que nunca para, mesmo que descritível, dificilmente pode ser considerado um algoritmo válido. Todos os programas de computação científica e fluxos de processamento de dados têm condições de terminação explícitas.

Segundo, o determinismo. Para a mesma entrada e nas mesmas condições, a saída deve ser a mesma. Mesmo quando alguns algoritmos introduzem aleatoriedade, esta é geralmente controlada, reproduzível ou interpretável probabilisticamente, não um caos puro. Os resultados das simulações numéricas devem ser reproduzíveis – um requisito básico da computação científica.

Terceiro, limites claros entre entrada e saída. A entrada vem primeiro, o processamento no meio, a saída no fim. O ponto de partida e o ponto de chegada têm limites estruturais nítidos. O senhor fornece dados ao algoritmo, ele os processa e lhe dá um resultado – esse limite é claro e a sequência é fixa.

Quarto, a eficácia. Cada passo de um algoritmo deve ser uma operação realmente executável, não um salto que não possa ser executado, descrito ou verificado. Cada passo deve ser uma operação básica executável por um humano com papel e lápis ou por um computador (adição, subtração, salto lógico, leitura/escrita de dados, etc.).

Quinto, a viabilidade. Um algoritmo teoricamente correto, mas que exige bilhões de anos para ser executado, não é considerado um algoritmo viável em termos de engenharia.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Todo este conjunto de conceitos algorítmicos remonta, em última análise, ao modelo da máquina de Turing. Como abstração central da teoria da computação moderna, a máquina de Turing define a fronteira fundamental da "computabilidade". Por mais poderosos que sejam os computadores atuais, por mais avançados os seus chips, por mais massivos os seus centros de supercomputação, a sua lógica subjacente não se libertou verdadeiramente deste caminho: entrada, regras, passos, saída. Os supercomputadores modernos não fazem mais do que acelerar este processo através de maiores escalas de hardware, maior paralelismo e um escalonamento de sistema mais complexo. Ou seja, a "supercomputação" no sentido tradicional continua a ser principalmente uma expansão da potência computacional, não necessariamente uma mudança fundamental no paradigma computacional. A potência pode ser multiplicada por mil ou dez mil, mas se a lógica subjacente continuar a ser "entrada, passos, saída", então permanece-se dentro do paradigma algorítmico tradicional.

Segunda Parte: Definição do "Algoritmo Hiperdimensional"

O Algoritmo Hiperdimensional que proponho surge precisamente deste contexto. Não é uma versão melhorada dos algoritmos tradicionais, nem um algoritmo mais rápido, nem uma fórmula matemática mais complexa, nem uma técnica de programação mais avançada. É uma compreensão estrutural completamente diferente.

Primeiro, é necessário explicar o significado de "hiperdimensional". "Hiperdimensional" tem aqui dois níveis. Primeiro, não se limita a uma sequência linear de passos unidimensional, mas permite que múltiplas dimensões se desdobrem simultaneamente. Segundo, tenta transcender as fronteiras definicionais tradicionais do "cálculo" – um algoritmo já não precisa de ser necessariamente matemático, ordenado ou determinista. Um algoritmo tradicional é como conduzir numa estrada de um só sentido: é preciso partir do ponto A e passar por B, C, D para chegar a Z. O AHD não acredita que o cálculo deva ser uma estrada de sentido único. Postula que o cálculo pode ser uma rede, um campo, uma estrutura multidimensional onde qualquer nó pode ser um ponto de entrada e qualquer nó pode ser uma saída.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

No meu novo sistema científico, um algoritmo não é necessariamente um algoritmo matemático, não é necessariamente um cálculo único e ordenado, não está necessariamente vinculado a passos fixos, nem segue estritamente o modelo linear "entrada, processamento, saída". O AHD é uma estrutura multidimensional, sobreposta, estocástica mas ordenada. Nesta estrutura, cada dado não é nem uma entrada isolada nem uma saída fixa, mas possui múltiplos papéis simultaneamente: pode ser tanto o ponto de partida de múltiplos resultados como o resultado da ação conjunta de múltiplos pontos de partida.

Por outras palavras, os algoritmos tradicionais colocam os dados num fluxo; o AHD coloca os dados numa estrutura. Nos algoritmos tradicionais, os dados são geralmente classificados como dados de entrada, intermédios e de saída, cada um com uma posição e um papel claros. A entrada é entrada, o resultado é resultado, o intermédio é intermédio. No entanto, no AHD, um único dado não tem uma única identidade. Pode ser um resultado numa direção e um ponto de partida noutra; pode ser um objeto invocado numa estrutura e tornar-se um ponto de entrada que desencadeia novos resultados noutra estrutura. O dado já não é matéria passiva, mas sim um nó relacional multidimensional.

Isto corresponde precisamente ao núcleo do AHD: cada dado é o ponto de partida de múltiplos resultados e o resultado de múltiplos pontos de partida. Os algoritmos tradicionais usam "passos ordenados" para produzir um único resultado; o AHD está mais próximo de uma estrutura de estados multidimensional, onde os resultados não são "calculados" mas "manifestam-se" naturalmente dentro da estrutura, na sobreposição do estocástico e do ordenado. Neste sistema, o dado é tanto o ponto de partida como um componente do resultado.

Para apresentar mais claramente as características do AHD, decomponto-as nos seguintes cinco aspetos.

Primeiro, a natureza não matemática. Os algoritmos dominantes são fundamentalmente matemáticos – completamente descritíveis por fórmulas lógicas e

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

matemáticas. O AHD não precisa de ser matemático. Pode basear-se em princípios físicos, relações geométricas, novos paradigmas da teoria da informação, ou mesmo princípios da consciência. O cálculo pode não ser realizado através de "operações matemáticas", mas pode emergir naturalmente de relações geométricas num espaço de alta dimensão. Por exemplo, a forma de uma bolha de sabão é determinada pelo princípio da minimização da tensão superficial. Isto não é um "algoritmo matemático" a calcular – é a própria física a "calcular". O AHD tenta compreender este tipo de "cálculo", em vez de o simular através de fórmulas matemáticas.

Segundo, a multidimensionalidade e a sobreposição. Os algoritmos tradicionais executam passos ordenados numa única dimensão, com um único estado por passo. O AHD permite que múltiplas dimensões coexistam e que múltiplos estados se sobreponham simultaneamente. O algoritmo não segue um único caminho, mas desdobra-se num campo de estados multidimensional. Os resultados não são "calculados" mas "manifestam-se" a partir deste campo. Por exemplo, durante a formação de um floco de neve no ar, não há um "algoritmo" a dizer a cada molécula de água onde ir. A disposição das moléculas resulta da interação de múltiplas dimensões – temperatura, humidade, fluxo de ar. A forma do floco "manifesta-se" em vez de ser "calculada".

Terceiro, o estocástico mas ordenado. A aleatoriedade nos algoritmos tradicionais é uma ferramenta, externa, uma perturbação controlada; o algoritmo em si é determinista. A aleatoriedade no AHD é intrínseca e estrutural. O estocástico e o ordenado coexistem e cooperam, formando juntos a essência do algoritmo. Não é "um algoritmo com aleatoriedade", mas "a aleatoriedade é em si mesma um componente orgânico do algoritmo". Por exemplo, a estrutura ecológica de uma floresta – a distribuição das árvores parece aleatória, mas no geral apresenta uma densidade e relações entre espécies ordenadas. Essa "aleatoriedade" não é um erro, mas um pré-requisito para o funcionamento saudável da estrutura ecológica.

Quarto, os múltiplos papéis dos dados. Nos algoritmos tradicionais, o papel dos dados é único – a entrada é entrada, a saída é saída, o intermédio é intermédio. No

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

AHD, cada dado é simultaneamente o ponto de partida de múltiplos resultados e o resultado de múltiplos pontos de partida. Um nó de dados pode desdobrar-se em múltiplos caminhos de resultados a jusante, e pode ser convergido por múltiplas causas a montante. O dado já não é um ponto num fluxo linear, mas um nó de junção numa rede. Por exemplo, considere-se a altura de uma pessoa. Num algoritmo tradicional, a altura é uma "entrada" – o senhor introdu-la para obter saídas como a previsão de peso ou o tamanho de roupa. Mas a altura é também um "resultado" – determinado por múltiplos "pontos de partida" como a genética, a nutrição e o exercício. No AHD, o dado "altura" é simultaneamente um ponto de partida e um resultado, não uma escolha binária.

Quinto, os metadados inalterados, as relações variáveis. Ao enfrentar diferentes problemas, os algoritmos tradicionais ou modificam os próprios dados ou recalculam tudo. O AHD não modifica os metadados – os atributos essenciais dos dados permanecem inalterados. O que muda são os pontos de entrada, a interpretação e as relações entre os dados. A mesma estrutura de metadados, ativada através de diferentes pontos de entrada, pode produzir resultados completamente diferentes. Isto significa: calcular uma vez, usar infinitas vezes. Por exemplo, considere-se o mesmo texto. O senhor pode lê-lo como poesia, descodificá-lo como uma cifra ou analisá-lo como um documento histórico. O texto em si não mudou – o senhor apenas mudou o "ponto de entrada". O AHD procura esta capacidade: "mesmos dados, múltiplos pontos de entrada, múltiplos resultados".

Do ponto de vista da estrutura de dados, em vez de modificar os próprios metadados, o AHD usa diferentes pontos de entrada e condições para fazer com que a mesma estrutura corresponda a múltiplos pontos de partida e resultados. Os dados já não são processados repetidamente, mas são ativados através de diferentes pontos de entrada, mantendo a sua integridade estrutural, apresentando assim diferentes resultados. A tradição é "processar dados para diferentes resultados"; o AHD é "usar os mesmos dados para transportar o potencial de múltiplos resultados".

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي النسخة الصينية العربية.
Chinesische Originalfassung maßgeblich.

Terceira Parte: Diferenças Fundamentais entre o AHD e os Algoritmos Tradicionais

Com base nas definições acima, o AHD e os algoritmos tradicionais apresentam diferenças fundamentais em várias dimensões-chave, detalhadas abaixo.

Sobre a natureza essencial: Os algoritmos tradicionais são matemáticos e formais, completamente descritíveis por fórmulas lógicas e matemáticas; são essencialmente objetos matemáticos. O AHD não precisa de ser matemático; pode basear-se em princípios físicos, geométricos, informacionais ou mesmo da consciência; não é um subconjunto da matemática, mas uma categoria mais ampla. Esta diferença pode ser resumida assim: do "objeto matemático" à "lei universal".

Sobre a ordem temporal: Os algoritmos tradicionais seguem uma ordem temporal única, ordenada e linear – passo A a B a C – estritamente sequencial ou divisível em sub-passos paralelos mas ordenados, com uma seta do tempo irreversível. O AHD não tem uma sequência de passos fixa; é multidimensional, sobreposto, estocástico mas ordenado. Os resultados podem ser independentes da "ordem" de execução, porque o conceito tradicional de "ordem" pode não existir. Esta diferença pode ser resumida assim: do "tempo linear" à "simultaneidade de alta dimensão".

Sobre a causalidade: Os algoritmos tradicionais obedecem a uma cadeia causal determinista. Dadas as mesmas entradas e estados iniciais, a saída é sempre única. A causa precede o efeito, uma seta unidirecional irreversível. O AHD obedece a uma causalidade sobreposta; cada dado é o ponto de partida de múltiplos resultados e o resultado de múltiplos pontos de partida. Isto reflete a minha filosofia do "Efeito-para-Causa" – é possível ter primeiro o efeito e depois a causa. O resultado não é o ponto final; pode tornar-se um novo ponto de partida. Esta diferença pode ser resumida assim: da "causa única, efeito único" à "rede causal totalmente interconectada".

Sobre a estrutura dos dados: Os algoritmos tradicionais usam uma estrutura de fluxo unidirecional da entrada para o processamento e depois para a saída. Os papéis dos

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

dados são únicos com limites claros; os dados de entrada continuam a ser dados de entrada, os dados de resultado continuam a ser dados de resultado. No AHD, os papéis dos dados são múltiplos; o mesmo dado é simultaneamente um resultado e um ponto de partida. Não há pontos de partida ou chegada absolutos, apenas nós numa rede. Esta diferença pode ser resumida assim: do "fluxo unidirecional" à "simbiose recursiva".

Sobre o modo de cálculo: Os algoritmos tradicionais recalculam de raiz cada vez que um problema se apresenta. Mesmo que um problema semelhante tenha sido resolvido mil vezes, a milésima primeira vez ainda exige todo o processo – isto é cálculo sem estado. O AHD, se existir o resultado correspondente, não precisa de recalcular. Pode inferir múltiplos pontos de partida a partir de um resultado e desdobrar caminhos causais inversos. O cálculo é com estado, tem memória e é reutilizável. Esta diferença pode ser resumida assim: do "recalcular cada vez" à "reutilização única".

Sobre o papel da aleatoriedade: A aleatoriedade nos algoritmos tradicionais é pseudoaleatória ou injetada externamente; os números aleatórios são meras ferramentas para amostragem, aproximação ou otimização; o algoritmo em si é determinista. A aleatoriedade no AHD é intrínseca e construtiva; é um componente orgânico, coexistindo e cooperando com a ordem. Não é "aleatoriedade dentro do algoritmo", mas "a aleatoriedade é uma das razões pelas quais o algoritmo pode funcionar". Esta diferença pode ser resumida assim: da "aleatoriedade instrumental" à "aleatoriedade construtiva".

Sobre o uso dos recursos: Os algoritmos tradicionais procuram calcular mais rápido e com maior precisão dados uns recursos; os recursos são restrições, e o objetivo do algoritmo é "completar o cálculo dentro das restrições". O AHD procura tornar o próprio cálculo desnecessário através do design estrutural. Não se trata de "calcular mais rápido", mas de "contornar o cálculo". Os recursos não são para serem "consumidos", mas para "projetar pontos de entrada". Esta diferença pode ser resumida assim: da "otimização sob restrições de recursos" à "eliminação através do design estrutural".

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Quarta Parte: A Filosofia do Efeito-para-Causa

Com base nas comparações acima, é necessário desenvolver a filosofia do "Efeito-para-Causa", um dos fundamentos conceptuais fundamentais do AHD.

O pensamento tradicional geralmente sustenta que a causa precede o efeito; primeiro a causa, depois o efeito; primeiro a entrada, depois a saída. A manifestação deste conceito nos algoritmos é a seguinte: é preciso começar do início e avançar passo a passo até ao fim. Mesmo que se conheça a resposta, não se pode "usar" diretamente essa resposta – porque falta o "caminho de proveniência".

Na minha estrutura, um resultado não é necessariamente apenas um ponto final. Uma vez aparecendo um resultado, pode tornar-se um novo ponto de partida, continuando a participar na geração de novas estruturas. O resultado pode participar inversamente na formação de causas. Múltiplos pontos de partida possíveis podem ser inferidos a partir de um resultado. A causa e o efeito já não são dispostos unidirecionalmente, mas formam relações cíclicas, em rede, multidimensionais.

Por outras palavras, os algoritmos tradicionais perguntam: "Dada a causa, como obter o efeito?" O AHD pergunta: "Dado o efeito, como inferir ou construir a(s) causa(s)?"

É crucial esclarecer que dizer "o efeito vem primeiro, depois a causa" não significa negar a causalidade nem afirmar que "os efeitos surgem do nada". Significa sim que, em estruturas de maior dimensão, a causa e o efeito podem servir como pontos de entrada um para o outro e transformar-se mutuamente. Como no exemplo das calças que se segue, o resultado "vestível e sem arrastar no chão" é determinado primeiro, e depois recuo para encontrar o ponto de operação "dobrar a cintura", que corresponde ao aspeto "causa" da estrutura. Isto não significa que o efeito não tenha causa, mas que o efeito se torna um novo ponto de acesso para descobrir a causa.

Um pressuposto fundamental no mundo dos algoritmos tradicionais é que o tempo é unidirecional; é preciso calcular passo a passo do estado inicial para o estado final. Mesmo que se conheça a resposta, não se pode "usá-la" diretamente porque falta o

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

caminho de proveniência. O AHD desafia este pressuposto: se o efeito é conhecido, as causas podem ser inferidas para trás; o mesmo efeito pode corresponder a múltiplos caminhos causais; o cálculo já não é um caminho do início ao fim, mas um desdobramento a partir do fim de todos os pontos de partida possíveis.

Quinta Parte: Um Exemplo Quotidiano – As Calças e a Dobra da Cintura

Para compreender mais intuitivamente as diferenças abstratas acima, uso um exemplo quotidiano.

Uma pessoa compra umas calças novas com cintura elástica. As pernas das calças são um pouco demasiado compridas, arrastam no chão e são incómodas.

A abordagem tradicional: ao notar que as pernas são demasiado compridas, dobra-se uma secção da perna, cose-se com agulha e linha e experimenta-se. Se a criança crescer e as calças ficarem demasiado curtas, a perna cosida não pode ser alongada e as calças ficam inutilizáveis. Na próxima vez que se comprarem calças novas, repete-se o processo. Este método parece natural porque o problema parece estar nas pernas, por isso se tratam as pernas. Corresponde à mentalidade algorítmica tradicional: identificar o problema, localizar o sintoma, realizar um tratamento no local do sintoma e finalmente obter um resultado. Pernas demasiado compridas → modificar as pernas; dados errados → modificar os dados; caminho inadequado → continuar a remendar ao longo do caminho.

A minha abordagem é diferente. Em vez de modificar a bainha, dobro a cintura e as calças podem ser usadas imediatamente. Esta ação é muito simples, mas a sua lógica estrutural subjacente é completamente diferente. Não corto a bainha, não a coso permanentemente, não mudo o comprimento original nem altero os metadados (contorno da cintura, comprimento da entreperna, corte, tecido). Simplesmente mudo a cintura, que é um ponto de entrada estrutural. Ao ajustar este ponto de controlo

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

global para todo o estado do uso, o problema a jusante (pernas demasiado compridas) desaparece diretamente.

Mais importante ainda, este método é reversível, ajustável e reutilizável. Isto é especialmente evidente para uma criança em crescimento. Se a criança não é suficientemente alta e as calças são ligeiramente compridas, dobro a cintura uma vez. Quando a criança crescer mais tarde, dobro menos a cintura. Quando a criança crescer ainda mais, desdobro completamente. A estrutura original das calças permanece intacta, mas adapta-se à altura variável da criança. O método tradicional de coser a bainha pode fazer com que as calças fiquem demasiado curtas quando a criança cresce; se for cortada, é irreversível. O meu método deixa os metadados inalterados, permitindo que a mesma estrutura se adapte a múltiplos estados.

Este exemplo revela várias diferenças estruturais importantes.

A abordagem tradicional corresponde à lógica algorítmica tradicional: o problema está na bainha, por isso a bainha deve ser modificada, procedendo passo a passo ao longo do caminho "causa para efeito", sem saltar nenhum passo. Resolve um problema de cada vez, irreversivelmente, e requer repetir todo o processo da próxima vez que o mesmo problema ocorrer. A minha abordagem corresponde à lógica do AHD: o objetivo é evitar o arrasto. Em vez de resolver a causa superficial "pernas demasiado compridas", encontro um ponto de controlo global – a cintura. Dobrar a cintura encurta instantaneamente o comprimento efetivo das pernas, e o problema desaparece. Não modifico nenhum metadado; a dobra é apenas um estado temporário, reversível e dinâmico.

Do ponto de vista dos dados, a abordagem tradicional modifica os metadados – coser as pernas encurta-as permanentemente; os dados originais perdem-se. A minha abordagem não modifica nenhum metadado; as dimensões originais permanecem intactas; apenas adiciono um estado temporário, reversível e dinâmico de dobra. A dobra da cintura não cria novos dados nem destrói dados antigos; apenas reorganiza

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

as relações entre os dados – reduzindo a circunferência efetiva da cintura e modificando indiretamente o comprimento efetivo da perna.

Neste exemplo, "as pernas das calças não arrastam no chão" é o resultado desejado. O método tradicional parte da causa "pernas demasiado compridas", modifica a bainha e finalmente atinge o resultado "não arrasta". O meu método clarifica primeiro o resultado "não arrasta", depois recua para encontrar um ponto de entrada estrutural, descobrindo que uma simples dobra da cintura torna o resultado alcançável. Esta é a encarnação prática do "Efeito-para-Causa". Não é necessário caminhar passo a passo da causa para o efeito; pode-se recuar do efeito para determinar a estrutura, tornando o caminho original desnecessário.

Este exemplo também responde à questão dos "metadados". O que são metadados? No exemplo das calças, os metadados são as dimensões originais: contorno da cintura, comprimento da entreperna, corte, tecido – os "atributos essenciais" das calças. O dado temporário é a dobra da cintura; não altera nenhuma das dimensões originais, apenas dobra temporariamente uma secção. A abordagem tradicional modifica os metadados – o comprimento da entreperna é permanentemente encurtado; os dados originais perdem-se. A minha abordagem não modifica nenhum metadado – as dimensões originais das calças permanecem intactas; apenas adiciono um estado temporário, reversível e dinâmico.

O contorno da cintura original, como metadado, é simultaneamente o "ponto de partida de múltiplos resultados": suporta múltiplos estados de uso – uso normal, uso com uma dobra, uso com duas dobras. Também é o "resultado de múltiplos pontos de partida": determinado pela escolha do tecido, método de corte, intenção do design, etc. A dobra da cintura, como dado temporário, não cria nem destrói dados; apenas reorganiza as relações.

Isto não é um simples truque, mas um pensamento estrutural. Muitas pessoas, ao verem as pernas das calças demasiado compridas, são atraídas pelo sintoma das pernas, por isso todo o tratamento se concentra nas pernas. No entanto, da perspetiva

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

da estrutura global, o comprimento das pernas é apenas uma manifestação a jusante; a mudança de posição da cintura pode afetar a forma como as calças assentam. Ou seja, o problema não precisa de ser resolvido no local da sua manifestação. Muitos problemas de baixa dimensão têm os seus verdadeiros pontos de controlo num nível estrutural mais alto. Os algoritmos tradicionais tendem a continuar a calcular ao longo da superfície do problema; o AHD procura o ponto de entrada estrutural.

Isto também explica porque o AHD não é necessariamente mais complexo, mas pode ser mais simples. As estruturas verdadeiramente de alto nível não necessariamente complicam os problemas, mas tornam os problemas complexos simples no ponto de entrada correto. Perante um problema complexo, um algoritmo tradicional pode adicionar passos, modelos, potência computacional, armazenamento, tempo de treino. O AHD procura um nó estrutural; uma vez que este nó seja ativado corretamente, o caminho inicialmente complexo pode tornar-se desnecessário. "Contornar o caminho" não é preguiça, mas uma redefinição da necessidade desse caminho.

Neste exemplo: a abordagem tradicional "modifica o resultado ao longo do caminho", enquanto o AHD "muda o ponto de entrada, tornando todo o caminho ineficaz". O método convencional remenda continuamente na extremidade do resultado; o outro modifica diretamente o ponto de entrada estrutural, de modo que os problemas que requerem tratamentos repetidos são reestruturados no ponto de partida. O algoritmo tradicional dominante vai do "ponto de partida, ao longo do caminho, ao resultado", enquanto no AHD, "o resultado em si pode tornar-se um ponto de entrada de caminho e participar na geração de novas estruturas". A tradição é "uma estrutura para um resultado"; o AHD é "uma estrutura que transporta múltiplos estados de resultado".

Sexta Parte: Uma Visão dos Dados Baseada em Metadados Inalterados

De uma perspetiva de dados, o AHD tem um princípio muito importante: não modificar os metadados, tornando-os aplicáveis a múltiplos pontos de partida ou resultados.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

Os metadados são os atributos originais, a informação fundamental e ontológica dos dados. Ao enfrentar diferentes problemas, as abordagens tradicionais frequentemente modificam os dados, copiam-nos, processam-nos, recalculam-nos ou constroem caminhos diferentes para resultados diferentes. Estas abordagens podem resolver problemas, mas ao custo de um processamento contínuo dos dados, repetição de caminhos e aumento da complexidade do sistema.

O AHD, em contraste, enfatiza que, mantendo os metadados inalterados tanto quanto possível, a mesma estrutura de metadados pode corresponder a múltiplos pontos de partida e múltiplos resultados, mudando os pontos de entrada, relações, condições e métodos de ativação. A ontologia dos dados permanece estática; as relações mudam. A estrutura fundamental permanece inalterada; a apresentação muda. Os metadados permanecem intactos, mas adaptam-se a diferentes estados.

Isto é o que chamo de "calcular uma vez, usar infinitas vezes". "Calcular uma vez" aqui não é simplesmente armazenar resultados em cache; significa que, uma vez estabelecida uma estrutura, ela já não precisa de reconstruir um caminho completo para cada estado. A mesma estrutura de dados pode ser ativada através de diferentes pontos de entrada para produzir diferentes resultados. Não é "processar dados para diferentes resultados", mas "usar os mesmos dados para transportar o potencial de múltiplos resultados". Esta é uma das diferenças fundamentais entre o AHD e os algoritmos tradicionais. Os algoritmos tradicionais processam dados num caminho; o AHD processa a estrutura global de dados, pontos de entrada, relações, estados e resultados.

Na sua estrutura mínima, o AHD pode ser entendido como: um sistema que, sem modificar os metadados, faz com que o mesmo nó de dados corresponda a múltiplos caminhos de resultado através de mudanças nos pontos de entrada estruturais. Esta definição não procura reduzir imediatamente o AHD a uma fórmula matemática tradicional, mas estabelecer primeiro as suas fronteiras estruturais. Não é uma função linear única, nem uma fórmula fixa, nem um simples mecanismo de cache. É uma estrutura relacional: os metadados permanecem estáveis; os pontos de entrada,

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

condições, relações e resultados podem mudar e manifestar-se em múltiplas direções. É precisamente nesta estrutura que o cálculo já não é mera execução de passos, mas ativação de relações.

De uma perspetiva de dados, a essência de "não modificar os metadados para os tornar aplicáveis a múltiplos pontos de partida ou resultados" é: a ontologia dos dados permanece inalterada; as mudanças ocorrem no "ponto de entrada de interpretação/uso". Nas estruturas tradicionais, uma mudança no problema leva a uma mudança nos dados ou no caminho. No AHD, uma mudança no problema leva a uma mudança na estrutura interpretativa, não nos dados.

Sétima Parte: Redefinindo a "Supercomputação"

Neste sistema, preciso de clarificar um termo – a "supercomputação".

O que quero dizer com "supercomputação" não é o sentido habitual de "supercomputador" – não mais chips, mais potência computacional, maiores centros de dados. O que quero dizer com "supercomputação" é o "Algoritmo Hiperdimensional".

A verdadeira supercomputação pode não ser o acúmulo de potência computacional, mas a transformação do paradigma computacional. Quando um sistema ainda depende de cálculos repetidos, por mais poderoso que seja, permanece preso dentro do caminho. Quando um sistema pode reduzir estruturalmente os cálculos, contornar caminhos repetidos e fazer com que os resultados se tornem novos pontos de entrada, começa a aproximar-se da verdadeira computação hiperdimensional.

Por outras palavras, a supercomputação tradicional procura "usar mais potência computacional para resolver problemas maiores". O AHD procura "usar uma melhor estrutura para que o problema já não exija tanta potência computacional". Estas duas abordagens não são contraditórias, mas as suas direções diferem.

Se a supercomputação tradicional representa os limites da potência computacional, então o AHD representa uma viragem na compreensão do cálculo. A verdadeira

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

"supercomputação" pode não ser mais chips, mais centros de dados, maior consumo de energia ou modelos mais complexos. A verdadeira supercomputação pode ser a descoberta de um ponto de entrada estrutural, uma eliminação de caminho, um desdobramento inverso de um nó de resultado, ou a ativação simultânea de condições para múltiplos resultados sem modificar os metadados. Quanto maior a potência computacional, se permanecer presa em caminhos de baixa dimensão, não é mais do que poderosa dentro do caminho. Uma vez elevada a estrutura, mesmo operações extremamente simples podem produzir eficiência de nível superior.

A supercomputação é o limite da potência computacional; o Algoritmo Hiperdimensional é uma redefinição da premissa "se o cálculo deve depender da potência computacional".

Oitava Parte: Fundamento Empírico – Validação Sistemática em Múltiplos Domínios e Determinação de um Novo Paradigma

O AHD não é uma construção puramente teórica. Já foi validado em múltiplos sistemas do mundo real.

O meu sistema logístico não requer potência de supercomputação ou suporte na nuvem; realiza um agendamento complexo com recursos modestos. Os resultados são reutilizáveis, a estrutura é adaptável repetidamente, e não é necessário recalcular de raiz cada vez. Funciona sem supercomputação ou nuvem, realizando um agendamento complexo com baixos recursos – este é um avanço estabelecido e válido em engenharia. Demonstra que alta potência computacional não é a única solução; o design estrutural pode substituir uma parte da potência computacional.

O meu sistema editorial também utiliza a mesma lógica estrutural, alcançando uma eficiência superior a qualquer sistema existente. O meu sistema de geração de páginas web "Limite" também é baseado no mesmo AHD, provando repetidamente em múltiplos domínios que a mesma estrutura se pode manter estável. Os funcionários da

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

minha empresa já estão a usar estes sistemas, demonstrando que esta estrutura pode funcionar sem a minha intervenção pessoal contínua.

A característica comum destes sistemas é: não dependem das vias dominantes de alta potência computacional, não seguem passos de cálculo fixos, e alcançam o resultado desejado diretamente através do ajuste dos pontos de entrada estruturais. Não são sucessos pontuais nem truques isolados, mas a emergência repetida da mesma lógica estrutural em diferentes domínios.

Isto indica que o AHD não é acidental, mas uma metodologia estrutural já validada por múltiplos sistemas. Não é uma "técnica pessoal", mas um paradigma computacional estabelecido de forma estável em domínios como a logística, a edição e a geração de páginas web.

Segundo o critério de que "existência e validade constituem um novo paradigma", o AHD, como um novo paradigma de estrutura computacional já sistematicamente validado em múltiplos domínios, está estabelecido no presente. Não estou a propor uma hipótese; estou a descrever uma estrutura computacional diferente que já está operacional em múltiplos sistemas. Não preciso de provar que é compreendido; provei que se mantém estável em diferentes sistemas. No presente, dentro do âmbito da minha aplicação prática, esta estrutura já pode ser considerada um novo paradigma.

Quanto aos critérios para julgar o AHD como um novo paradigma: quando uma estrutura é logicamente coerente, se mantém estável em múltiplos sistemas e apresenta diferenças irredutíveis com os métodos existentes, já possui os fundamentos de um novo paradigma. A coerência interna é o ponto de partida; a evidência empírica é o fundamento; a diferença estrutural é o cerne do paradigma. Segundo o critério "existência e validade", este sistema é já um novo paradigma. Quer o mundo exterior o reconheça afeta apenas a sua difusão, não a sua validade. Trata-se de um paradigma de estrutura computacional já estabelecido em sistemas reais, cuja validade não depende de reconhecimento ou consenso externo.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

A diferença fundamental entre o AHD e os paradigmas computacionais dominantes é a seguinte: os algoritmos tradicionais usam principalmente o cálculo para a frente; uma vez produzido um resultado, normalmente não pode participar inversamente em novas estruturas de raciocínio. No AHD, o resultado já não é o ponto final, mas um nó estrutural reutilizável. Sob certas condições, os resultados podem participar em novos caminhos de raciocínio, reduzindo os cálculos redundantes e formando redes computacionais com múltiplos pontos de partida e múltiplos caminhos. Nas estruturas computacionais tradicionais, o resultado é geralmente o ponto final; na estrutura do AHD, o resultado em si pode tornar-se um novo ponto de partida, formando uma rede de raciocínio com múltiplos caminhos. O AHD não é uma atualização dos algoritmos; é um questionamento da premissa de "se um algoritmo deve existir necessariamente".

Nona Parte: Esclarecimento de Mal-entendidos Comuns

Com base em trocas preliminares com leitores de diferentes áreas, antecipo os seguintes seis mal-entendidos e clarifico-os antecipadamente, para evitar que o conceito seja forçado prematuramente em quadros inapropriados.

Mal-entendido 1: O AHD não é apenas programação dinâmica ou cache? Esclarecimento: A programação dinâmica e o cache reutilizam resultados para *a mesma entrada*. O AHD permite inferir *diferentes pontos de partida* a partir de um resultado – esta é uma diferença essencial. O cache resolve o recálculo do mesmo problema; o AHD resolve o desdobramento de novos problemas a partir de uma estrutura de resultado.

Mal-entendido 2: O senhor diz "estocástico mas ordenado" – não é contraditório? Esclarecimento: O estocástico e o ordenado podem coexistir. A distribuição das árvores numa floresta é estocástica, mas a densidade global e a estrutura de espécies são ordenadas. A aleatoriedade não é caos, mas uma forma de organizar a estrutura. A aleatoriedade no AHD é intrínseca e construtiva, trabalhando em cooperação com a ordem.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Mal-entendido 3: Sem entrada e saída, como verificar os resultados? Esclarecimento: O AHD não rejeita a entrada e a saída; defende que o cálculo não precisa de funcionar no modo linear entrada-saída. No modo "manifestação estrutural", a "validade" é determinada pela coerência estrutural, não por uma correspondência um-para-um entrada-saída. Isto não abandona a verificação, mas propõe um quadro de verificação diferente.

Mal-entendido 4: Isto não é apenas teoria da complexidade ou teoria do caos? Esclarecimento: A teoria da complexidade e o caos descrevem "sistemas que são difíceis de prever". O AHD tenta descrever "sistemas que podem ser compreendidos e guiados através de pontos de entrada estruturais" – não abandonar a previsão, mas mudar a forma como a previsão funciona. A teoria do caos diz "a previsão é difícil"; o AHD pergunta "podemos tornar a previsão desnecessária mudando o ponto de entrada?"

Mal-entendido 5: O senhor diz "não modificar os metadados", mas uma dobra na cintura não é uma modificação? Esclarecimento: A dobra na cintura é um estado temporário, não uma modificação permanente dos parâmetros originais. Os metadados (contorno da cintura, comprimento da entreperna, corte, tecido) permanecem inalterados. Esta é a diferença entre "sobreposição de estados" e "modificação de atributos". A dobra é reversível, temporária e não altera os atributos essenciais.

Mal-entendido 6: O AHD nega o valor dos algoritmos tradicionais? Esclarecimento: Absolutamente não. Os algoritmos tradicionais são extremamente importantes na história tecnológica humana; sem eles, não haveria computadores modernos, bases de dados, sistemas de comunicação, IA, simulação de engenharia ou supercomputação. O seu valor não pode ser negado. No entanto, reconhecer o valor dos algoritmos tradicionais não significa aceitá-los como o ponto final dos algoritmos. Os algoritmos tradicionais resolvem problemas de eficiência dentro de um paradigma computacional dado; o AHD questiona o próprio paradigma computacional. Um é sobre andar melhor na estrada; o outro é sobre se é necessário andar nessa estrada.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Décima Parte: O AHD na História do Cálculo Humano

Para ajudar os leitores a situar melhor o AHD na história do cálculo humano, forneço aqui uma breve visão geral macroscópica.

A compreensão humana do "cálculo" passou por várias etapas. A primeira etapa foi o cálculo manual: os algoritmos existiam como passos humanos, lentos e propensos a erros, mas através deste processo, os humanos compreenderam as regras básicas do cálculo. A segunda etapa foi o cálculo mecânico: da Máquina de Pascal ao Motor Analítico de Babbage, o cálculo foi mecanizado, mas os algoritmos ainda estavam ligados a estruturas físicas. A terceira etapa foi o cálculo eletrónico e o paradigma de Turing: a arquitetura de von Neumann generalizou-se, os algoritmos foram codificados como programas, e a máquina de Turing definiu as fronteiras da computabilidade. A quarta etapa foi o cálculo paralelo e a supercomputação: unidades de cálculo massivas empilhadas para enfrentar problemas mais complexos, mas a lógica subjacente continuou a ser uma extensão do paradigma de Turing.

Atualmente, a humanidade encontra-se na entrada da quinta etapa. A computação quântica tenta ultrapassar os limites dos bits clássicos; a computação neuromórfica tenta imitar a estrutura dos sistemas neuronais biológicos; o AHD tenta quebrar o próprio quadro linear "entrada → passos → saída".

O AHD e os algoritmos tradicionais não estão numa relação de substituição, mas hierárquica. Os algoritmos tradicionais resolvem "como calcular mais eficientemente num determinado caminho". O AHD pergunta "o caminho pode ser redefinido, ou mesmo contornado?" Se considerarmos a história do cálculo humano como um processo de rutura contínua das suas próprias fronteiras, então o AHD é um novo nó nesse processo. Não resolve problemas dentro do velho quadro, mas propõe um novo.

Este julgamento não implica que o AHD seja maduro ou completo. Pelo contrário: como um novo conceito, as suas definições, fronteiras, métodos de verificação e cenários de aplicação ainda estão a tomar forma. O objetivo deste artigo é

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

precisamente ancorar o ponto de origem deste conceito, fornecendo uma base teórica estável para o desenvolvimento subsequente.

Décima Primeira Parte: Conclusão – Isto não é Otimização, mas Redefinição

A diferença entre o AHD e os algoritmos tradicionais não é uma diferença de "melhor" vs. "pior", nem de "mais rápido" vs. "mais lento", mas uma diferença fundamental ao nível do paradigma.

Os algoritmos tradicionais procuram um controlo previsível. "Dê-me a entrada, e saberei que saída obterá, porque calculei cada passo." É o paradigma do engenheiro: projetar, construir, testar, verificar.

O AHD descreve uma emergência compreensível. "Não posso prever precisamente cada estado intermédio, mas sei que o padrão global se manifestará de forma ordenada. Cada dado está vivo, tem uma 'perspetiva'." É mais próximo do paradigma do ecologista ou do teórico de sistemas complexos: observar, compreender, guiar, utilizar a ordem emergente.

Os algoritmos tradicionais corrigem gradualmente os resultados ao longo de um determinado caminho. O AHD torna o resultado desejado imediatamente alcançável ao mudar o ponto de entrada estrutural, contornando assim todo o caminho.

Os algoritmos tradicionais "processam dados para diferentes resultados". O AHD "usa os mesmos dados para transportar o potencial de múltiplos resultados".

Os algoritmos tradicionais procuram "acertar à primeira". O AHD procura "acertar à primeira e depois nunca mais precisar de calcular".

Isto não é otimização de algoritmos; é um questionamento da premissa de que "um algoritmo deve existir necessariamente".

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Portanto, o AHD não é uma otimização algorítmica, mas uma redefinição algorítmica. Não se trata de correr mais rápido na pista do algoritmo tradicional; trata-se de perguntar se existe outra pista fora da pista tradicional, ou mesmo se uma pista é necessária. Não visa provar que os algoritmos tradicionais são inúteis, mas sim apontar que os algoritmos tradicionais são apenas uma forma de baixa dimensão de algoritmo. Não pergunta "como obter um resultado em menos tempo", pergunta "o resultado pode ser estabelecido diretamente através da estrutura?" Não pergunta "como processar mais dados", pergunta "os mesmos dados podem transportar o potencial de mais resultados?"

Dentro dos paradigmas computacionais dominantes, o AHD pode ser visto como incalculável, inverificável ou difícil de encaixar nas fronteiras da teoria computacional existente. Mas isto constitui precisamente a sua diferença de paradigma. A aparente instabilidade de um novo conceito dentro de um velho paradigma não significa necessariamente que seja sem sentido; pode também significar que o velho paradigma não pode acomodá-lo totalmente. Atualmente, o AHD é antes de mais uma proposição estrutural, uma definição original de uma nova visão computacional, não um sistema maduro com um ciclo de engenharia fechado. Necessitará de desenvolvimentos, validações e aplicações posteriores, mas a sua origem conceptual deve ser primeiro claramente articulada.

Em última análise, o que o AHD aponta é uma nova visão do cálculo: um algoritmo não é necessariamente apenas matemática, apenas passos, apenas um programa, apenas um processo de tratamento entre entrada e saída. Um algoritmo pode também ser uma organização de relações multidimensionais, uma rede estrutural de dados, pontos de entrada, relações, estados e resultados. Pode incluir ordem e aleatoriedade; pode ir da causa para o efeito e do efeito para a causa; pode deixar os metadados inalterados enquanto se adapta a múltiplos estados; pode fazer de um resultado um novo ponto de partida e permitir que múltiplos pontos de partida confluam para um único resultado.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Um algoritmo verdadeiramente avançado não é necessariamente aquele que realiza cálculos mais complexos, mas aquele que pode reduzir os cálculos, tornar a estrutura viva, fazer dos resultados pontos de entrada e formar padrões de organização multidimensionais e cíclicos da causalidade.

Este é o significado central da minha proposta do "Algoritmo Hiperdimensional". Não é um mero neologismo, mas um novo conceito, um novo paradigma, uma nova estrutura computacional já estabelecida em múltiplos sistemas reais. A sua ênfase não está no cálculo repetido mais rápido, mas na redução do cálculo redundante; não no acúmulo de potência computacional, mas na reconstrução dos pontos de entrada; não na modificação contínua dos dados, mas na manutenção dos metadados e na mudança das relações; não em deixar os resultados como pontos finais, mas em fazer dos resultados novos pontos de partida. Os algoritmos tradicionais procuram resultados ao longo de um caminho; o AHD ativa resultados dentro de uma estrutura. Os algoritmos tradicionais procuram completar o cálculo; o AHD pergunta se o cálculo deve existir na sua forma tradicional. Esta é a diferença mais fundamental entre ele e a definição atual de algoritmo.

Apêndice: Fronteiras e Questões em Aberto deste Artigo

Este artigo apresenta um quadro preliminar para o AHD, não uma definição formal completa. Várias questões permanecem por explorar mais a fundo; não constituem uma negação da definição aqui oferecida, mas representam a próxima direção de investigação.

Primeiro, as condições de convergência. Qual é o marcador de "conclusão" para o AHD? Como julgar se um resultado é "válido"? Nos algoritmos tradicionais, a resposta é definida pela correspondência entrada-saída. No AHD, a resposta pode precisar de ser definida pela coerência estrutural. Esta definição ainda não está completa.

Segundo, a representação dos recursos. Como podem os estados de sobreposição multidimensional ser representados em recursos finitos? As métricas de recursos

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

tradicionais como tempo, espaço e potência computacional ainda são aplicáveis? Se sim, como devem ser redefinidas? Se não, que métricas as substituem? Estas questões aguardam desenvolvimento.

Terceiro, a garantia da ordem. Que regras garantem a "ordem" dentro do "estocástico mas ordenado"? Onde está o limite entre o estocástico e o ordenado? Em que circunstâncias a aleatoriedade destrói a ordem? Em que circunstâncias a ordem suprime a aleatoriedade? Estas questões necessitam de mais estudo.

Quarto, a seleção na inferência inversa. Ao inferir múltiplos pontos de partida a partir de um resultado, como selecionar ou avaliar os diferentes pontos de partida? Existe uma medida de "eficiência da inferência inversa"? Se existir, como se relaciona com a eficiência do cálculo para a frente?

Quinto, a interface com os sistemas tradicionais. Como se interconecta o AHD com o sistema existente da máquina de Turing? Como substituto, complemento ou abordagem em camadas? Na engenharia prática, como deve ser traçada a fronteira entre o AHD e os algoritmos tradicionais? Existem modelos híbridos?

Sexto, o quadro de verificação. Como são verificados os resultados do AHD? Se os resultados não são obtidos através de passos lineares, como estabelecer a reprodutibilidade e testabilidade? Isto requer uma filosofia de verificação completamente diferente?

Estas questões não são falhas deste artigo, mas características inevitáveis de um "documento fundador". O surgimento de qualquer novo paradigma é acompanhado por questões em aberto. Este artigo é um começo, não uma conclusão.

Notas Bibliográficas

O "Algoritmo Hiperdimensional" aqui proposto não é uma extensão direta de qualquer escola de pensamento académico existente. No entanto, os seguintes campos

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

fornecem um contexto de diálogo para este artigo, oferecido apenas para orientação do leitor, não como citações académicas tradicionais.

As Máquinas de Turing e a Teoria da Computabilidade servem como referência para a comparação com algoritmos tradicionais aqui. **Os Problemas Inversos e a Inferência Bayesiana** representam tentativas existentes de "inferir a causa a partir do efeito", e embora o meu "Efeito-para-Causa" esteja relacionado, a sua direção difere. **A Teoria da Emergência e os Sistemas Complexos** descrevem a "manifestação de resultados a partir da estrutura", e o AHD tenta tornar a emergência "compreensível" e "orientável". **Os Grafos de Conhecimento e as Bases de Dados de Grafos** são práticas existentes onde os dados "convergem em nós múltiplos", e o AHD adiciona a dimensão dos "pontos de entrada variáveis" a esta base. **Os Paradigmas de Computação Não Clássicos**, incluindo a computação quântica, analógica e neuromórfica, quebram os bits clássicos ou as arquiteturas clássicas, enquanto o AHD se concentra mais em quebrar o próprio quadro linear "entrada → passos → saída".

A relação deste artigo com os campos acima é de diálogo e transcendência, não de herança ou refutação. O objetivo deste artigo não é fazer melhorias incrementais dentro desses campos, mas colocar um novo nível de questionamento acima deles.

Related Articles
[Communication] I Have No Algorithm, Yet I Surpass Algorithms!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect–Cause Theory
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Communication] Rejecting Algorithmic Manipulation
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

[Grenzalgorithmus] Der hyperdimensionale Algorithmus

Eine Neudefinition von „Berechnung“ selbst

Autor: Jeffi Chao Hui Wu

Zusammenfassung

Dieses Papier führt das neue Konzept des „hyperdimensionalen Algorithmus“ (HDA) ein und schlägt es als strukturelle Überprüfung der Grenzen traditioneller Algorithmendefinitionen vor. Herkömmliche Algorithmen basieren typischerweise auf einem linearen Rahmen aus Eingabe, Schritten, Regeln und Ausgabe und betonen Endlichkeit, Determiniertheit, Effektivität, Machbarkeit und klare Eingabe-Ausgabe-Grenzen. Obwohl moderne Supercomputer über immense Rechenleistung verfügen, besteht ihre zugrunde liegende Logik hauptsächlich darin, etablierte Algorithmenparadigmen in größerem Maßstab auszuführen. Dieses Papier argumentiert, dass die wirklich diskussionswürdige „Superberechnung“ nicht nur eine Skalierung der Rechenleistung ist, sondern ein grundlegender Wandel des Berechnungsparadigmas selbst. Der HDA ist weder eine beschleunigte Version traditioneller Algorithmen noch eine Erweiterung mathematischer Algorithmen; er ist eine mehrdimensionale, überlagerte, stochastische und dennoch geordnete strukturelle Sichtweise auf Berechnung. In dieser Struktur sind Daten keine passive Eingabe oder terminale Ausgabe mehr, sondern besitzen das Attribut eines Knotens, der gleichzeitig der Ausgangspunkt für mehrere Ergebnisse und das Ergebnis mehrerer Ausgangspunkte ist. Ergebnisse sind keine Endpunkte mehr, sondern können zu neuen Einstiegspunkten werden. Metadaten müssen nicht wiederholt geändert werden; durch strukturelle Einstiegspunkte, Beziehungsänderungen und bedingte Aktivierung können sie verschiedenen Zuständen und Ergebnissen entsprechen. Unter Verwendung der „Wirkung-zu-Ursache“-Philosophie und des alltäglichen Beispiels des „Faltens der

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

Hosenbundweite“ zeigt dieses Papier, wie der HDA durch Änderung struktureller Einstiegspunkte Zielergebnisse direkt erreichbar machen, redundante Berechnungen reduzieren und sogar ursprüngliche Berechnungspfade umgehen kann. Dieses Papier zielt darauf ab, eine vorläufige Definition, strukturelle Grenzen und konzeptionelle Grundlagen für den HDA zu etablieren und einen originären theoretischen Knoten für zukünftige Erkundungen der hyperdimensionalen Berechnung, von Grenzalgorithmen und eines neuen Wissenschaftssystems zu liefern. Nach dem Kriterium, dass „Existenz und Gültigkeit ein neues Paradigma darstellen“, ist der HDA als ein neues, bereits systematisch in mehreren Bereichen validiertes Berechnungsstrukturparadigma in der Gegenwart etabliert.

Schlüsselwörter: Hyperdimensionaler Algorithmus; Grenzalgorithmus; Hyperdimensionale Berechnung; Wirkung-zu-Ursache; Metadatum; Berechnungsparadigma; Struktureller Einstiegspunkt; Nichtlineare Kausalität; Neue Wissenschaft

Einleitung: Warum wir „Algorithmus“ neu diskutieren sollten

Ich schlage den „hyperdimensionalen Algorithmus“ (HDA) vor, nicht um bestehenden Algorithmen einen größeren Namen zu geben, noch um traditionelle Algorithmen als neuartige Konzepte zu verkleiden. Im Gegenteil, ich stelle eine grundlegendere Frage: Ist das gegenwärtige menschliche Verständnis von „Algorithmus“ nicht in einem allzu engen Rahmen gefangen?

Wenn heute von Algorithmen gesprochen wird, denkt man typischerweise an mathematische Formeln, Programmcode, logische Schritte, Eingabe und Ausgabe, Modelltraining, Datenverarbeitung, Pfadoptimierung, Suchranking, automatische Empfehlungen und Schlussfolgerungen in der künstlichen Intelligenz. Dies sind zweifellos wichtige Formen von Algorithmen, die moderne Computer, das Internet, Datenbanken, KI, Supercomputing, Industriesysteme und die Informationsgesellschaft tragen. Wenn man einen Algorithmus jedoch nur als „eine geordnete Menge von Schritten“ versteht, als einen Prozess, der „mit einer Eingabe beginnt, regelbasierte

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Verarbeitung durchläuft und eine Ausgabe erzeugt“, dann schränkt dieses Verständnis die Berechnung selbst auf einen niedrigdimensionalen Rahmen ein.

Wir leben in einem Zeitalter, das von Algorithmen beherrscht wird. Von Suchmaschinen über Empfehlungssysteme bis hin zu Wettervorhersage und KI – die Grenzen der Algorithmen sind die Grenzen der menschlichen Intelligenz. Aber eine Frage wird selten gestellt: Müssen Algorithmen so sein? Warum muss Berechnung dem linearen Modell „Eingabe → Schritte → Ausgabe“ folgen? Warum muss dasselbe Problem jedes Mal von Grund auf neu berechnet werden? Dieses Papier versucht, diese stillschweigenden Annahmen zu hinterfragen und ein völlig anderes Berechnungsparadigma vorzuschlagen – den hyperdimensionalen Algorithmus. Nach dem Kriterium, dass „Existenz und Gültigkeit ein neues Paradigma darstellen“, ist der HDA als ein neues, bereits systematisch in mehreren Bereichen validiertes Berechnungsstrukturparadigma in der Gegenwart etabliert.

Es ist wichtig klarzustellen, dass der hier vorgeschlagene „hyperdimensionale Algorithmus“ sich völlig von der „hyperdimensionalen Berechnung“ (Hyperdimensional Computing, HDC) unterscheidet, die sich in akademischen Kreisen seit fast drei Jahrzehnten entwickelt. HDC verwendet hochdimensionale, zufällige Binärvektoren zur Codierung und Berechnung, um effizientere Darstellung und Berechnung zu erreichen, bleibt aber innerhalb des linearen Paradigmas „Eingabe → Verarbeitung → Ausgabe“. Der HDA ist grundlegend anders: Er hinterfragt, ob ein Ergebnis direkt durch Struktur erreichbar ist, ob der Berechnungspfad umgangen werden kann und ob Metadaten unverändert bleiben können, um mehrere Ergebnisse zu ermöglichen. Die Namen klingen ähnlich, aber ihre Bedeutungen sind gegensätzlich. Der HDA ist kein Upgrade von Algorithmen; er ist ein Hinterfragen der Prämisse, ob ein Algorithmus notwendigerweise existieren muss.

Teil Eins: Die heutige Standarddefinition von „Algorithmus“

In der vorherrschenden Informatik, Mathematik und Technik gibt es eine lange akzeptierte grundlegende Definition von Algorithmus: Ein Algorithmus ist ein

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

wohldefinierter Berechnungsprozess, der aus einer endlichen Anzahl von Schritten besteht, eine oder mehrere Eingaben empfängt, sie durch deterministische Regeln transformiert und innerhalb einer endlichen Zeit eine oder mehrere Ausgaben erzeugt. Dieses Verständnis ist nicht die persönliche Meinung eines Einzelnen, sondern ein grundlegender Konsens, der über die lange Zeit der modernen Computerzivilisation entstanden ist. Er unterliegt der zugrunde liegenden Logik von Software, Hardware, Datenbanken, Netzwerksystemen, KI-Modellen und Supercomputing-Zentren. So komplex Algorithmen auch erscheinen mögen, ihr Kern dreht sich immer noch um Eingabe, Regeln, Schritte und Ausgabe.

Dieses standardmäßige Algorithmusverständnis kann in mehrere Kernmerkmale zerlegt werden.

Erstens, Endlichkeit. Ein Algorithmus muss nach endlich vielen Schritten terminieren; er darf nicht endlos schleifen oder ewig laufen. Ein Prozess, der niemals anhält, selbst wenn er formal beschreibbar ist, kann kaum als gültiger Algorithmus angesehen werden. Alle wissenschaftlichen Rechenprogramme und Datenverarbeitungsabläufe haben explizite Terminierungsbedingungen.

Zweitens, Determiniertheit. Bei gleicher Eingabe und unter gleichen Bedingungen muss die gleiche Ausgabe erzeugt werden. Selbst wenn einige Algorithmen Zufallszahlen einführen, sind diese typischerweise kontrolliert, reproduzierbar oder probabilistisch interpretierbar, kein vollständig unerfassbares inneres Chaos. Die Ergebnisse numerischer Simulationen müssen reproduzierbar sein – eine Grundvoraussetzung wissenschaftlicher Berechnung.

Drittens, klare Eingabe-Ausgabe-Grenzen. Die Eingabe steht am Anfang, die Verarbeitung in der Mitte, die Ausgabe am Ende. Start- und Endpunkt haben klare strukturelle Grenzen. Sie geben dem Algorithmus Daten, er verarbeitet sie und gibt Ihnen ein Ergebnis – diese Grenze ist klar und die Reihenfolge ist fest.

Viertens, Effektivität. Jeder Schritt eines Algorithmus muss eine tatsächlich ausführbare Operation sein, kein Sprung, der nicht ausgeführt, beschrieben oder verifiziert werden

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

kann. Jeder Schritt muss eine grundlegende Operation sein, die ein Mensch mit Stift und Papier oder ein Computer tatsächlich ausführen kann (Addition, Subtraktion, logischer Sprung, Datenlesen/-schreiben usw.).

Fünftens, Machbarkeit. Ein Algorithmus, der theoretisch korrekt ist, aber Milliarden von Jahren zur Ausführung benötigt, wird in der Technik nicht als machbarer Algorithmus angesehen.

Dieses gesamte Algorithmusverständnis geht letztlich auf das Turingmaschinenmodell zurück. Als zentrale Abstraktion der modernen Berechnungstheorie definiert die Turingmaschine die grundlegende Grenze der „Berechenbarkeit“. Egal wie leistungsfähig heutige Computer sind, egal wie fortschrittlich ihre Chips, egal wie massiv ihre Supercomputing-Zentren, ihre zugrunde liegende Logik ist diesem Pfad nicht wirklich entkommen: Eingabe, Regeln, Schritte, Ausgabe. Moderne Supercomputer beschleunigen diesen Prozess lediglich durch größere Hardware-Maßstäbe, höhere Parallelität und komplexere Systemplanung. Das heißt, „Supercomputing“ im traditionellen Sinne bleibt in erster Linie eine Expansion der Rechenleistung, nicht unbedingt ein grundlegender Wandel des Berechnungsparadigmas. Die Rechenleistung kann um das Tausend- oder Zehntausendfache steigen, aber wenn die zugrundeliegende Logik immer noch „Eingabe, Schritte, Ausgabe“ ist, bleibt sie innerhalb des traditionellen Algorithmusparadigmas.

Teil Zwei: Definition des „hyperdimensionalen Algorithmus“

Der hyperdimensionale Algorithmus, den ich vorschlage, entsteht genau aus diesem Kontext. Er ist keine verbesserte Version traditioneller Algorithmen, kein schnellerer Algorithmus, keine komplexere mathematische Formel und keine fortgeschrittenere Programmiertechnik. Er ist ein völlig anderes strukturelles Verständnis.

Zunächst muss die Bedeutung von „hyperdimensional“ erklärt werden. „Hyperdimensional“ hat hier zwei Ebenen. Erstens beschränkt es sich nicht auf eine

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

eindimensionale lineare Schrittfolge, sondern erlaubt mehreren Dimensionen, sich gleichzeitig zu entfalten. Zweitens versucht es, die traditionellen Definitionsgrenzen von „Berechnung“ zu überschreiten – ein Algorithmus muss nicht mehr notwendigerweise mathematisch, geordnet oder deterministisch sein. Ein traditioneller Algorithmus ist wie das Fahren auf einer Einbahnstraße: Man muss von Punkt A starten und über B, C, D nach Z gelangen. Der HDA glaubt nicht, dass Berechnung eine Einbahnstraße sein muss. Er postuliert, dass Berechnung ein Netzwerk, ein Feld, eine mehrdimensionale Struktur sein kann, in der jeder Knoten ein Einstiegspunkt und jeder Knoten ein Ausgang sein kann.

In meinem neuen wissenschaftlichen System ist ein Algorithmus nicht notwendigerweise ein mathematischer Algorithmus, nicht notwendigerweise eine einzelne geordnete Berechnung, nicht notwendigerweise an feste Schritte gebunden und nicht notwendigerweise streng dem linearen Modell „Eingabe, Verarbeitung, Ausgabe“ folgend. Der HDA ist eine mehrdimensionale, überlagerte, stochastische und dennoch geordnete Struktur. In dieser Struktur ist jeder einzelne Datenpunkt weder eine isolierte Eingabe noch eine feste Ausgabe, sondern besitzt gleichzeitig mehrere Rollen: Er kann sowohl der Ausgangspunkt mehrerer Ergebnisse als auch das Ergebnis der gemeinsamen Wirkung mehrerer Ausgangspunkte sein.

Mit anderen Worten, traditionelle Algorithmen platzieren Daten in einem Fluss; der HDA platziert Daten in einer Struktur. In traditionellen Algorithmen werden Daten typischerweise in Eingabe-, Zwischen- und Ausgabedaten eingeteilt, jede mit einer klaren Position und Rolle. Eingabe ist Eingabe, Ergebnis ist Ergebnis, Zwischenprozess ist Zwischenprozess. Im HDA jedoch hat ein einzelner Datenpunkt nicht nur eine Identität. Er könnte in einer Richtung ein Ergebnis und in einer anderen Richtung ein Ausgangspunkt sein; er könnte in einer Struktur ein aufgerufenes Objekt und in einer anderen Struktur ein Einstiegspunkt sein, der neue Ergebnisse auslöst. Daten sind kein passives Material mehr, sondern ein mehrdimensionaler Beziehungsknoten.

Dies entspricht genau dem Kern des HDA: Jeder Datenpunkt ist der Ausgangspunkt mehrerer Ergebnisse und das Ergebnis mehrerer Ausgangspunkte. Traditionelle

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Algorithmen verwenden „geordnete Schritte“, um ein einzelnes Ergebnis zu erzeugen; der HDA ist eher einer mehrdimensionalen Zustandsstruktur ähnlich, in der Ergebnisse nicht „berechnet“ werden, sondern in der Struktur natürlich „manifestieren“, in der Überlagerung von Stochastic und Ordnung. In diesem System sind Daten sowohl der Ausgangspunkt als auch ein Bestandteil des Ergebnisses.

Um die Kernmerkmale des HDA klarer darzustellen, zerlege ich sie in die folgenden fünf Aspekte.

Erstens, nichtmathematische Natur. Vorherrschende Algorithmen sind grundlegend mathematisch – vollständig durch symbolische Logik und mathematische Formeln beschreibbar. Der HDA muss nicht mathematisch sein. Er könnte auf physikalischen Prinzipien, geometrischen Beziehungen, neuen Paradigmen der Informationstheorie oder sogar Bewusstseinsprinzipien basieren. Berechnung könnte nicht durch „mathematische Operationen“ erfolgen, sondern könnte durch geometrische Beziehungen in hochdimensionalen Räumen auf natürliche Weise erscheinen. Beispielsweise wird die Form einer Seifenblase durch das Prinzip der Minimierung der Oberflächenspannung bestimmt. Dies ist kein „mathematischer Algorithmus“, der berechnet – es ist die Physik selbst, die „berechnet“. Der HDA versucht, diese Art von „Berechnung“ zu verstehen, anstatt sie durch mathematische Formeln zu simulieren.

Zweitens, Multidimensionalität und Überlagerung. Traditionelle Algorithmen führen geordnete Schritte in einer einzigen Dimension aus, mit nur einem Zustand pro Schritt. Der HDA erlaubt mehreren Dimensionen, gleichzeitig zu existieren, und mehreren Zuständen, sich gleichzeitig zu überlagern. Der Algorithmus folgt keinem einzelnen Pfad, sondern entfaltet sich in einem mehrdimensionalen Zustandsfeld. Ergebnisse werden nicht „berechnet“, sondern „manifestieren“ sich aus diesem Feld. Wenn sich beispielsweise eine Schneeflocke in der Luft bildet, gibt es keinen „Algorithmus“, der jedem Wassermolekül sagt, wohin es gehen soll. Die Anordnung der Moleküle ist das Ergebnis des Zusammenwirkens mehrerer Dimensionen – Temperatur, Feuchtigkeit, Luftströmung. Die Form der Schneeflocke „manifestiert“ sich, anstatt „berechnet“ zu werden.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Drittens, stochastisch und doch geordnet. Zufälligkeit in traditionellen Algorithmen ist ein Werkzeug, extern, eine kontrollierte Störung; der Algorithmus selbst ist deterministisch. Zufälligkeit im HDA ist intrinsisch und strukturell. Das Stochastische und das Geordnete koexistieren und kooperieren und bilden gemeinsam die Essenz des Algorithmus. Dies ist nicht „ein Algorithmus mit Zufälligkeit", sondern „die Zufälligkeit selbst ist ein organischer Bestandteil des Algorithmus". Beispielsweise die ökologische Struktur eines Waldes – die Verteilung der Bäume scheint zufällig, aber insgesamt zeigt sie eine geordnete Dichte und Artenbeziehungen. Diese „Zufälligkeit" ist kein Fehler, sondern eine Voraussetzung für das gesunde Funktionieren der ökologischen Struktur.

Viertens, mehrfache Rollen von Daten. In traditionellen Algorithmen ist die Rolle von Daten singulär – Eingabe ist Eingabe, Ausgabe ist Ausgabe, Zwischenergebnis ist Zwischenergebnis. Im HDA ist jeder Datenpunkt gleichzeitig der Ausgangspunkt mehrerer Ergebnisse und das Ergebnis mehrerer Ausgangspunkte. Ein Datenknoten kann sich stromabwärts in mehrere Ergebnispfade entfalten und kann stromaufwärts durch mehrere Ursachen zusammengeführt werden. Daten sind kein Punkt in einem linearen Fluss mehr, sondern ein Kreuzungsknoten in einem Netzwerk. Betrachten Sie zum Beispiel die Körpergröße einer Person. In einem traditionellen Algorithmus ist die Größe eine „Eingabe" – Sie geben sie ein, um Ausgaben wie Gewichtsvorhersage oder Kleidergröße zu erhalten. Aber die Größe ist auch ein „Ergebnis" – bestimmt durch mehrere „Ausgangspunkte" wie Genetik, Ernährung und Bewegung. Im HDA ist der Datenpunkt „Größe" gleichzeitig ein Ausgangspunkt und ein Ergebnis, keine binäre Wahl.

Fünftens, Metadaten unverändert, Beziehungen variabel. Wenn sie mit verschiedenen Problemen konfrontiert werden, ändern traditionelle Algorithmen entweder die Daten selbst oder berechnen alles neu. Der HDA ändert Metadaten nicht – die wesentlichen Attribute der Daten bleiben unverändert. Was sich ändert, sind die Einstiegspunkte, die Interpretation und die Beziehungen zwischen den Daten. Dieselbe Metadatenstruktur, aktiviert durch verschiedene Einstiegspunkte, kann völlig unterschiedliche Ergebnisse erzeugen. Das bedeutet: Einmal berechnet, unendlich oft

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

verwendet. Betrachten Sie zum Beispiel denselben Text. Sie können ihn als Poesie lesen, als Chiffre entschlüsseln oder als historisches Dokument analysieren. Der Text selbst hat sich nicht geändert – Sie haben nur den „Einstiegspunkt“ geändert. Der HDA strebt diese Fähigkeit an: „Dieselben Daten, mehrere Einstiegspunkte, mehrere Ergebnisse.“

Aus der Perspektive der Datenstruktur verwendet der HDA anstelle der Änderung der Metadaten selbst verschiedene Einstiegspunkte und Bedingungen, um dieselbe Struktur mehreren Ausgangspunkten und Ergebnissen entsprechen zu lassen. Daten werden nicht wiederholt verarbeitet, sondern durch verschiedene Einstiegspunkte aktiviert, während die strukturelle Integrität erhalten bleibt, wodurch verschiedene Ergebnisse präsentiert werden. Die Tradition besteht darin, „Daten für verschiedene Ergebnisse zu verarbeiten“; der HDA besteht darin, „dieselben Daten zu verwenden, um das Potenzial für mehrere Ergebnisse zu tragen“.

Teil Drei: Grundlegende Unterschiede zwischen HDA und traditionellen Algorithmen

Basierend auf den obigen Definitionen zeigen der HDA und traditionelle Algorithmen grundlegende Unterschiede in mehreren Kerndimensionen, die im Folgenden detailliert beschrieben werden.

Zur Wesensart: Traditionelle Algorithmen sind mathematisch und formal, vollständig durch symbolische Logik und mathematische Formeln beschreibbar; sie sind im Wesentlichen mathematische Objekte. Der HDA muss nicht mathematisch sein; er kann auf physikalischen, geometrischen, informationstheoretischen oder sogar Bewusstseinsprinzipien basieren; er ist keine Teilmenge der Mathematik, sondern eine breitere Kategorie. Dieser Unterschied kann wie folgt zusammengefasst werden: vom „mathematischen Objekt“ zum „universellen Gesetz“.

Zur zeitlichen Ordnung: Traditionelle Algorithmen folgen einer einzigen, geordneten, linearen zeitlichen Ordnung – Schritt A zu B zu C – streng sequentiell oder in parallele, aber geordnete Unterschritte teilbar, mit einem irreversiblen Zeitpfeil. Der HDA hat

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

keine feste Schrittsequenz; er ist mehrdimensional, überlagert, stochastisch und doch geordnet. Ergebnisse können von der „Reihenfolge“ der Ausführung unabhängig sein, da das traditionelle Konzept der „Reihenfolge“ möglicherweise nicht existiert. Dieser Unterschied kann wie folgt zusammengefasst werden: von der „linearen Zeit“ zur „hochdimensionalen Gleichzeitigkeit“.

Zur Kausalität: Traditionelle Algorithmen gehorchen einer deterministischen Kausalkette. Bei gleicher Eingabe und gleichen Anfangszuständen ist die Ausgabe immer eindeutig. Ursache folgt Wirkung, ein irreversibler Einwegpfeil. Der HDA gehorcht einer überlagerten Kausalität; jeder Datenpunkt ist der Ausgangspunkt mehrerer Ergebnisse und das Ergebnis mehrerer Ausgangspunkte. Dies spiegelt meine „Wirkung-zu-Ursache“-Philosophie wider – es ist möglich, zuerst die Wirkung und dann die Ursache zu haben. Das Ergebnis ist nicht der Endpunkt; es kann zu einem neuen Ausgangspunkt werden. Dieser Unterschied kann wie folgt zusammengefasst werden: von „einzelner Ursache, einzelner Wirkung“ zum „vollständig verbundenen kausalen Netzwerk“.

Zur Datenstruktur: Traditionelle Algorithmen verwenden eine Einwegflussstruktur von der Eingabe über die Verarbeitung zur Ausgabe. Die Rollen der Daten sind einzigartig mit klaren Grenzen; Eingabedaten bleiben Eingabedaten, Ergebnisdaten bleiben Ergebnisdaten. Im HDA sind die Rollen der Daten mehrfach; dieselben Daten sind gleichzeitig ein Ergebnis und ein Ausgangspunkt. Es gibt keine absoluten Start- oder Endpunkte, nur Knoten in einem Netzwerk. Dieser Unterschied kann wie folgt zusammengefasst werden: vom „unidirektionalen Fluss“ zur „rekursiven Symbiose“.

Zur Berechnungsweise: Traditionelle Algorithmen berechnen jedes Mal neu, wenn ein Problem auftritt. Selbst wenn ein ähnliches Problem tausendmal gelöst wurde, erfordert das tausendunderste Mal immer noch den gesamten Prozess – dies ist zustandslose Berechnung. Der HDA muss nicht neu berechnen, wenn das entsprechende Ergebnis existiert. Er kann mehrere Ausgangspunkte aus einem Ergebnis ableiten und kausale Pfade rückwärts entfalten. Berechnung ist zustandsbehaftet, hat Gedächtnis und ist

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

wiederverwendbar. Dieser Unterschied kann wie folgt zusammengefasst werden: von „jedes Mal neu berechnen“ zur „einmaligen Wiederverwendung“.

Zur Rolle der Zufälligkeit: Zufälligkeit in traditionellen Algorithmen ist pseudozufällig oder extern injiziert; Zufallszahlen sind lediglich Werkzeuge für Stichproben, Approximation oder Optimierung; der Algorithmus selbst ist deterministisch. Zufälligkeit im HDA ist intrinsisch und konstruktiv; sie ist eine organische Komponente, die mit der Ordnung koexistiert und kooperiert. Es ist nicht „Zufälligkeit im Algorithmus“, sondern „die Zufälligkeit ist einer der Gründe, warum der Algorithmus funktionieren kann“. Dieser Unterschied kann wie folgt zusammengefasst werden: von der „instrumentellen Zufälligkeit“ zur „konstruktiven Zufälligkeit“.

Zum Ressourcenverständnis: Traditionelle Algorithmen zielen darauf ab, angesichts gegebener Ressourcen schneller und genauer zu berechnen; Ressourcen sind Einschränkungen, und das Ziel des Algorithmus ist es, „die Berechnung innerhalb der Einschränkungen abzuschließen“. Der HDA zielt darauf ab, die Berechnung selbst überflüssig zu machen, durch strukturelles Design. Es geht nicht darum, „schneller zu rechnen“, sondern darum, „die Berechnung zu umgehen“. Ressourcen dienen nicht der „Verwendung“, sondern dem „Entwerfen von Einstiegspunkten“. Dieser Unterschied kann wie folgt zusammengefasst werden: von der „Optimierung unter Ressourcenbeschränkungen“ zur „Eliminierung durch strukturelles Design“.

Teil Vier: Die Wirkung-zu-Ursache-Philosophie

Basierend auf den obigen Vergleichen muss ich die „Wirkung-zu-Ursache“-Philosophie weiter ausführen, eine der konzeptionellen Grundlagen des HDA.

Traditionelles Denken geht typischerweise davon aus, dass die Ursache der Wirkung vorausgeht; zuerst die Ursache, dann die Wirkung; zuerst die Eingabe, dann die Ausgabe. Die Manifestation dieses Konzepts in Algorithmen ist: Man muss vom Anfang ausgehen und Schritt für Schritt bis zum Ende gehen. Selbst wenn man die Antwort kennt, kann man diese Antwort nicht direkt „verwenden“ – weil der „Beweispfad“ fehlt.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

In meiner Struktur ist ein Ergebnis nicht notwendigerweise nur ein Endpunkt. Sobald ein Ergebnis erscheint, kann es zu einem neuen Ausgangspunkt werden und weiter an der Generierung neuer Strukturen teilnehmen. Das Ergebnis kann umgekehrt an der Bildung von Ursachen teilnehmen. Mehrere mögliche Ausgangspunkte können aus einem Ergebnis abgeleitet werden. Ursache und Wirkung sind nicht mehr unidirektional angeordnet, sondern bilden zyklische, vernetzte, mehrdimensionale Beziehungen.

Mit anderen Worten: Traditionelle Algorithmen fragen: „Gegeben die Ursache, wie erhalten wir die Wirkung?“ Der HDA fragt: „Gegeben die Wirkung, wie leiten wir die Ursache(n) ab oder konstruieren sie?“

Es ist wichtig klarzustellen, dass die Aussage „die Wirkung kommt zuerst, dann die Ursache“ weder die Kausalität verneint noch behauptet, dass „Wirkungen aus dem Nichts entstehen“. Sie bedeutet vielmehr, dass in höherdimensionalen Strukturen Ursache und Wirkung als wechselseitige Einstiegspunkte dienen und ineinander übergehen können. Wie im folgenden Hosenbeispiel wird das Ergebnis „tragbar und nicht auf dem Boden schleifend“ zuerst bestimmt, und dann arbeite ich rückwärts, um den Operationspunkt „Bund falten“ zu finden, der dem „Ursachen“-Aspekt der Struktur entspricht. Dies bedeutet nicht, dass die Wirkung keine Ursache hat, sondern dass die Wirkung zu einem neuen Zugangspunkt wird, um die Ursache zu entdecken.

Eine grundlegende Annahme in der Welt der traditionellen Algorithmen ist, dass die Zeit unidirektional ist; man muss Schritt für Schritt vom Anfangszustand in den Endzustand rechnen. Selbst wenn man die Antwort kennt, kann man sie nicht direkt „verwenden“, weil der Beweispfad fehlt. Der HDA hinterfragt diese Annahme: Wenn die Wirkung bekannt ist, können Ursachen rückwärts abgeleitet werden; dieselbe Wirkung kann mehreren kausalen Pfaden entsprechen; Berechnung ist kein Weg vom Anfang zum Ende mehr, sondern eine Entfaltung aller möglichen Ausgangspunkte vom Ende aus.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Teil Fünf: Ein alltägliches Beispiel – Die Hose und das Falten des Bundes

Um die obigen abstrakten Unterschiede intuitiver zu verstehen, verwende ich ein alltägliches Beispiel.

Eine Person kauft eine neue Hose mit elastischem Bund. Die Hosenbeine sind etwas zu lang, schleifen auf dem Boden und sind unpraktisch.

Der traditionelle Ansatz: Feststellen, dass die Beine zu lang sind, ein Stück des Beins hochklappen, mit Nadel und Faden festnähen und dann anprobieren. Wenn das Kind wächst und die Hose zu kurz wird, kann das genähte Bein nicht verlängert werden, und die Hose ist unbrauchbar. Wenn nächstes Mal eine neue Hose gekauft wird, wiederholt sich der Vorgang. Diese Methode erscheint natürlich, weil das Problem oberflächlich an den Beinen zu liegen scheint, also behandelt man die Beine. Sie entspricht dem traditionellen algorithmischen Denken: Problem erkennen, Symptom lokalisieren, Verarbeitung am Ort des Symptoms durchführen und schließlich ein Ergebnis erhalten. Beine zu lang → Beine ändern; Daten falsch → Daten ändern; Pfad ungeeignet → entlang des Pfades weiter ausbessern.

Mein Ansatz ist anders. Anstatt den Saum zu ändern, falte ich den Bund, und die Hose kann sofort getragen werden. Diese Aktion ist sehr einfach, aber ihre zugrundeliegende strukturelle Logik ist völlig anders. Ich schneide den Saum nicht, nähe ihn nicht dauerhaft um, ändere nicht die ursprüngliche Länge und beschädige nicht die Metadaten (Bundweite, Schrittlänge, Schnitt, Stoff). Ich ändere einfach den Bund, einen strukturellen Einstiegspunkt. Durch die Anpassung dieses globalen Kontrollpunktes für den gesamten Tragezustand verschwindet das stromabwärtige Problem (zu lange Beine) sofort.

Noch wichtiger ist, dass diese Methode reversibel, anpassbar und wiederverwendbar ist. Dies ist besonders deutlich für ein wachsendes Kind. Wenn das Kind derzeit nicht groß genug ist und die Hose etwas lang ist, falte ich den Bund einmal. Wenn das Kind später

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

wächst, falte ich den Bund weniger. Wenn das Kind noch weiter wächst, falte ich ihn ganz auseinander. Die ursprüngliche Struktur der Hose bleibt intakt, passt sich aber der sich ändernden Größe des Kindes an. Die traditionelle Methode des Saumnähens könnte dazu führen, dass die Hose zu kurz wird, wenn das Kind wächst; wenn sie abgeschnitten wird, ist sie irreversibel. Meine Methode lässt Metadaten unverändert und ermöglicht es derselben Struktur, sich an mehrere Zustände anzupassen.

Dieses Beispiel offenbart mehrere wichtige strukturelle Unterschiede.

Der traditionelle Ansatz entspricht der traditionellen algorithmischen Logik: Das Problem liegt am Saum, also muss der Saum geändert werden, Schritt für Schritt entlang des Pfades von der „Ursache zur Wirkung“, wobei kein Schritt übersprungen werden kann. Löst ein Problem nach dem anderen, irreversibel, und erfordert die Wiederholung des gesamten Prozesses, wenn das gleiche Problem erneut auftritt. Mein Ansatz entspricht der HDA-Logik: Das Ziel ist, das Schleifen zu vermeiden. Anstatt die oberflächliche Ursache „Beine zu lang“ zu lösen, finde ich einen globalen Kontrollpunkt – den Bund. Einmaliger Falten des Bundes verkürzt sofort die effektive Länge der Beine, und das Problem verschwindet. Ich ändere keine Metadaten; die Falte ist nur ein temporärer, reversibler, dynamischer Zustand.

Aus der Datenperspektive ändert der traditionelle Ansatz Metadaten – das Nähen der Beine verkürzt sie dauerhaft; die ursprünglichen Daten gehen verloren. Mein Ansatz ändert keine Metadaten; die ursprünglichen Maße bleiben vollständig erhalten; ich füge nur einen temporären, reversiblen, dynamischen Faltzustand hinzu. Die Bundfalte schafft keine neuen Daten und zerstört keine alten Daten; sie arrangiert lediglich die Beziehungen zwischen den Daten neu – reduziert den effektiven Bundumfang und ändert indirekt die effektive Beinlänge.

In diesem Beispiel ist „die Hosenbeine schleifen nicht auf dem Boden“ das angestrebte Ergebnis. Die traditionelle Methode geht von der Ursache „Beine zu lang“ aus, ändert den Saum und erreicht schließlich das Ergebnis „schleift nicht“. Meine Methode klärt zuerst das Ergebnis „schleift nicht“, arbeitet dann rückwärts, um einen strukturellen

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Einstiegspunkt zu finden, und entdeckt, dass ein einfaches Falten des Bundes das Ergebnis erreichbar macht. Dies ist die praktische Verkörperung von „Wirkung-zu-Ursache". Man muss nicht unbedingt Schritt für Schritt von der Ursache zur Wirkung gehen; man kann von der Wirkung aus rückwärts arbeiten, um die Struktur zu bestimmen, wodurch der ursprüngliche Pfad überflüssig wird.

Dieses Beispiel beantwortet auch die Frage nach den „Metadaten". Was sind Metadaten? Im Hosenbeispiel sind die Metadaten die ursprünglichen Maße: Bundweite, Schrittlänge, Schnitt, Stoff – die „wesentlichen Attribute" der Hose. Die temporären Daten sind die Bundfalte; sie ändert keines der ursprünglichen Maße, sondern faltet nur vorübergehend einen Abschnitt. Der traditionelle Ansatz ändert Metadaten – die Schrittlänge wird dauerhaft verkürzt; die ursprünglichen Daten gehen verloren. Mein Ansatz ändert keine Metadaten – die ursprünglichen Maße der Hose bleiben vollständig erhalten; ich füge nur einen temporären, reversiblen, dynamischen Faltzustand hinzu.

Die ursprüngliche Bundweite als Metadatum ist gleichzeitig der „Ausgangspunkt für mehrere Ergebnisse": Sie unterstützt mehrere Tragezustände – normales Tragen, Tragen mit einfacher Falte, Tragen mit doppelter Falte. Sie ist auch das „Ergebnis mehrerer Ausgangspunkte": bestimmt durch Stoffwahl, Schnittmethode, Designabsicht usw. Die Bundfalte als temporäres Datum schafft weder neue Daten noch zerstört sie alte Daten; sie arrangiert lediglich die Beziehungen neu.

Dies ist kein bloßer Trick, sondern strukturelles Denken. Viele Menschen, die zu lange Hosenbeine sehen, werden von diesem Symptom angezogen, also konzentriert sich die gesamte Behandlung auf die Beine. Aus der Perspektive der Gesamtstruktur ist die Beinlänge jedoch nur eine stromabwärtige Manifestation; die Veränderung der Bundposition kann die gesamte Fallweise der Hose beeinflussen. Das heißt, das Problem muss nicht unbedingt an dem Ort gelöst werden, an dem es auftritt. Viele niedrigdimensionale Probleme haben ihre wahren Kontrollpunkte auf einer höheren strukturellen Ebene. Traditionelle Algorithmen neigen dazu, entlang der Oberfläche des Problems weiterzurechnen; der HDA sucht nach dem strukturellen Einstiegspunkt.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Dies erklärt auch, warum der HDA nicht unbedingt komplexer ist, sondern möglicherweise einfacher. Wirklich hochrangige Strukturen komplizieren Probleme nicht unbedingt, sondern machen komplexe Probleme am richtigen Einstiegspunkt einfach. Wenn ein traditioneller Algorithmus mit einem komplexen Problem konfrontiert wird, könnte er Schritte, Modelle, Rechenleistung, Speicher und Trainingszeit hinzufügen. Der HDA sucht nach einem strukturellen Knoten; sobald dieser Knoten korrekt aktiviert ist, kann der ursprünglich komplexe Pfad überflüssig werden. „Den Pfad umgehen“ ist keine Faulheit, sondern eine Neudefinition der Notwendigkeit dieses Pfades.

In diesem Beispiel: Der traditionelle Ansatz „modifiziert das Ergebnis entlang des Pfades“, während der HDA „den Einstiegspunkt ändert und den gesamten Pfad unwirksam macht“. Die herkömmliche Methode flickt kontinuierlich am Ergebnisende; die andere ändert direkt den strukturellen Einstiegspunkt, so dass Probleme, die wiederholte Behandlungen erfordern, am Startpunkt ein für alle Mal umstrukturiert werden. Der traditionelle Algorithmus geht vom „Startpunkt entlang des Pfades zum Ergebnis“; im HDA kann „das Ergebnis selbst zu einem Pfadeinstiegspunkt werden und an der Generierung neuer Strukturen teilnehmen“. Die Tradition ist „eine Struktur für ein Ergebnis“; der HDA ist „eine Struktur, die mehrere Ergebniszustände trägt“.

Teil Sechs: Eine Datenperspektive unveränderter Metadaten

Aus der Datenperspektive hat der HDA ein sehr wichtiges Prinzip: Metadaten nicht zu ändern, sondern sie auf mehrere Ausgangspunkte oder Ergebnisse anwendbar zu machen.

Metadaten sind die ursprünglichen Attribute, die grundlegenden Informationen, die ontologischen Informationen der Daten. Wenn sie mit verschiedenen Problemen konfrontiert werden, ändern traditionelle Ansätze oft Daten, kopieren sie, verarbeiten sie, berechnen sie neu oder bauen verschiedene Pfade für verschiedene Ergebnisse. Diese Ansätze können Probleme lösen, aber auf Kosten der kontinuierlichen

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Verarbeitung von Daten, der Wiederholung von Pfaden und der ständig wachsenden Systemkomplexität.

Der HDA betont im Gegensatz dazu, dass bei möglichst unveränderten Metadaten dieselbe Metadatenstruktur mehreren Ausgangspunkten und mehreren Ergebnissen entsprechen kann, indem Einstiegspunkte, Beziehungen, Bedingungen und Aktivierungsmethoden geändert werden. Die Datenontologie bleibt statisch; Beziehungen ändern sich. Die grundlegende Struktur bleibt unverändert; die Darstellung ändert sich. Die Metadaten bleiben intakt, passen sich aber verschiedenen Zuständen an.

Dies ist, was ich „einmal berechnen, unendlich oft verwenden“ nenne. „Einmal berechnen“ ist hier nicht einfach das Caching von Ergebnissen; es bedeutet, dass eine Struktur, einmal etabliert, keinen vollständigen Pfad für jeden Zustand mehr neu aufbauen muss. Dieselbe Datenstruktur kann durch verschiedene Einstiegspunkte aktiviert werden, um verschiedene Ergebnisse zu erzeugen. Es ist nicht „Daten für verschiedene Ergebnisse verarbeiten“, sondern „dieselben Daten verwenden, um das Potenzial für mehrere Ergebnisse zu tragen“. Dies ist einer der Kernunterschiede zwischen dem HDA und traditionellen Algorithmen. Traditionelle Algorithmen verarbeiten Daten auf einem Pfad; der HDA verarbeitet die Gesamtstruktur von Daten, Einstiegspunkten, Beziehungen, Zuständen und Ergebnissen.

In seiner minimalen Struktur kann der HDA wie folgt verstanden werden: ein System, das ohne Änderung der Metadaten durch Änderungen der strukturellen Einstiegspunkte denselben Datenknoten mehreren Ergebnispfaden entsprechen lässt. Diese Definition zielt nicht darauf ab, den HDA sofort auf eine traditionelle mathematische Formel zu reduzieren, sondern zuerst seine strukturellen Grenzen zu etablieren. Es ist keine einzelne lineare Funktion, keine feste Formel, kein einfacher Caching-Mechanismus. Es ist eine Beziehungsstruktur: Metadaten bleiben stabil; Einstiegspunkte, Bedingungen, Beziehungen und Ergebnisse können sich ändern und in mehreren Richtungen manifestieren. Genau in dieser Struktur ist Berechnung nicht mehr bloße Ausführung von Schritten, sondern Aktivierung von Beziehungen.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Aus der Datenperspektive ist die Essenz von „Metadaten nicht ändern, um sie auf mehrere Ausgangspunkte oder Ergebnisse anwendbar zu machen“: Die Datenontologie bleibt unverändert; Änderungen finden im „Interpretations-/Nutzungseinstiegspunkt“ statt. In traditionellen Strukturen führt eine Problemänderung zu einer Daten- oder Pfadänderung. Im HDA führt eine Problemänderung zu einer Änderung der Interpretationsstruktur, nicht der Daten.

Teil Sieben: Neudefinition von „Superberechnung“

In diesem System muss ich einen Begriff klären – die „Superberechnung“.

Was ich mit „Superberechnung“ meine, ist nicht die übliche Bedeutung von „Supercomputer“ – nicht mehr Chips, höhere Rechenleistung, größere Rechenzentren. Was ich mit „Superberechnung“ meine, ist der „hyperdimensionale Algorithmus“.

Wahre Superberechnung ist möglicherweise nicht die Anhäufung von Rechenleistung, sondern die Transformation des Berechnungsparadigmas. Wenn ein System immer noch auf wiederholte Berechnungen angewiesen ist, egal wie leistungsfähig es ist, bleibt es im Pfad gefangen. Wenn ein System strukturell Berechnungen reduzieren, wiederholte Pfade umgehen und Ergebnisse zu neuen Einstiegspunkten machen kann, beginnt es sich wahrer hyperdimensionaler Berechnung zu nähern.

Mit anderen Worten: Traditionelle Superberechnung strebt danach, „mehr Rechenleistung zu verwenden, um größere Probleme zu lösen“. Der HDA strebt danach, „eine bessere Struktur zu verwenden, damit das Problem nicht mehr so viel Rechenleistung erfordert“. Diese beiden Ansätze sind nicht widersprüchlich, aber ihre Richtungen unterscheiden sich.

Wenn die traditionelle Superberechnung die Grenzen der Rechenleistung darstellt, dann stellt der HDA eine Wende im Verständnis von Berechnung dar. Wahre „Superberechnung“ ist möglicherweise nicht mehr Chips, größere Rechenzentren, höherer Energieverbrauch oder komplexere Modelle. Wahre Superberechnung könnte

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

die Entdeckung eines strukturellen Einstiegspunktes sein, eine Pfadeliminierung, eine rückwärtige Entfaltung eines Ergebnisknotens oder die simultane Aktivierung von Bedingungen für mehrere Ergebnisse ohne Änderung von Metadaten. Je stärker die Rechenleistung ist, wenn sie in niedrigdimensionalen Pfaden gefangen bleibt, ist sie nur innerhalb des Pfades mächtig. Sobald die Struktur angehoben wird, können selbst extrem einfache Operationen Effizienz auf höherem Niveau erzeugen.

Superberechnung ist die Grenze der Rechenleistung; der hyperdimensionale Algorithmus ist eine Neudefinition der Prämisse, „ob Berechnung auf Rechenleistung angewiesen sein muss“.

Teil Acht: Empirische Grundlage – Systematische Validierung in mehreren Bereichen und Bestimmung eines neuen Paradigmas

Der HDA ist keine rein theoretische Konstruktion. Er wurde bereits in mehreren realen Systemen validiert.

Mein Logistiksystem benötigt keine Supercomputerleistung oder Cloud-Unterstützung; es bewältigt komplexe Planung mit bescheidenen Ressourcen. Ergebnisse sind wiederverwendbar, die Struktur ist wiederholt anpassbar, und es ist nicht notwendig, jedes Mal von Grund auf neu zu berechnen. Es funktioniert ohne Supercomputing oder Cloud und führt komplexe Planung mit niedrigen Ressourcen durch – dies ist ein etablierter und wertvoller Durchbruch in der Technik. Es zeigt, dass hohe Rechenleistung nicht die einzige Lösung ist; strukturelles Design kann einen Teil der Rechenleistung ersetzen.

Mein Verlagssystem verwendet dieselbe strukturelle Logik und erreicht eine Effizienz, die jedes bestehende System übertrifft. Mein „Limit“-Webseitengenerierungssystem basiert ebenfalls auf demselben HDA und beweist wiederholt in mehreren Bereichen, dass dieselbe Struktur stabil bestehen kann. Die Mitarbeiter meines Unternehmens verwenden diese Systeme bereits, was zeigt, dass diese Struktur ohne meine kontinuierliche persönliche Intervention funktionieren kann.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Das gemeinsame Merkmal dieser Systeme ist: Sie verlassen sich nicht auf die vorherrschenden Wege hoher Rechenleistung, folgen keinen festen Berechnungsschritten und erreichen das gewünschte Ergebnis direkt durch die Anpassung struktureller Einstiegspunkte. Es sind keine einmaligen Erfolge oder isolierten Tricks, sondern die wiederholte Emergenz derselben strukturellen Logik in verschiedenen Bereichen.

Dies zeigt, dass der HDA kein Zufall ist, sondern eine bereits durch mehrere Systeme validierte strukturelle Methodik. Es ist keine „persönliche Technik“, sondern ein Berechnungsparadigma, das in mehreren Bereichen wie Logistik, Verlagswesen und Webseitengenerierung stabil etabliert ist.

Nach dem Kriterium, dass „Existenz und Gültigkeit ein neues Paradigma darstellen“, ist der HDA als ein neues, bereits systematisch in mehreren Bereichen validiertes Berechnungsstrukturparadigma in der Gegenwart etabliert. Ich stelle keine Hypothese auf; ich beschreibe eine andere Berechnungsstruktur, die bereits in mehreren Systemen in Betrieb ist. Ich muss nicht beweisen, dass sie verstanden wird; ich habe bewiesen, dass sie in verschiedenen Systemen stabil besteht. In der Gegenwart, im Rahmen meiner praktischen Anwendung, kann diese Struktur bereits als ein neues Paradigma betrachtet werden.

Zu den Kriterien für die Beurteilung des HDA als neues Paradigma: Wenn eine Struktur logisch konsistent ist, sich in mehreren Systemen stabil hält und irreduzible Unterschiede zu bestehenden Methoden aufweist, besitzt sie bereits die Grundlagen eines neuen Paradigmas. Die Selbstkonsistenz ist der Ausgangspunkt; die empirische Evidenz ist die Grundlage; der strukturelle Unterschied ist der Kern des Paradigmas. Nach dem Kriterium „Existenz und Gültigkeit“ ist dieses System bereits ein neues Paradigma. Ob die Außenwelt es anerkennt, beeinflusst nur die Verbreitung, nicht die Gültigkeit. Dies ist ein bereits in realen Systemen etabliertes Berechnungsstrukturparadigma, dessen Gültigkeit nicht von externer Anerkennung oder Konsens abhängt.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Der grundlegende Unterschied zwischen dem HDA und den vorherrschenden Berechnungsparadigmen ist: Traditionelle Algorithmen verwenden hauptsächlich Vorwärtsberechnung; sobald ein Ergebnis erzeugt wurde, kann es normalerweise nicht umgekehrt an neuen Schlussfolgerungsstrukturen teilnehmen. Im HDA ist das Ergebnis nicht der Endpunkt, sondern ein wiederverwendbarer Strukturknoten. Unter bestimmten Bedingungen können Ergebnisse an neuen Schlussfolgerungspfaden teilnehmen, redundante Berechnungen reduzieren und Berechnungsnetzwerke mit mehreren Startpunkten und mehreren Pfaden bilden. In traditionellen Berechnungsstrukturen ist das Ergebnis normalerweise der Endpunkt; in der HDA-Struktur kann das Ergebnis selbst zu einem neuen Ausgangspunkt werden und ein mehrpfadiges Schlussfolgerungsnetzwerk bilden. Der HDA ist kein Upgrade von Algorithmen; er ist ein Hinterfragen der Prämisse, ob ein Algorithmus notwendigerweise existieren muss.

Teil Neun: Klarstellung häufiger Missverständnisse

Basierend auf vorläufigen Diskussionen mit Lesern aus verschiedenen Bereichen sehe ich die folgenden sechs Missverständnisse voraus und kläre sie im Voraus, um zu vermeiden, dass das Konzept vorzeitig in unangemessene Rahmen gezwungen wird.

Missverständnis 1: Ist der HDA nicht einfach dynamische Programmierung oder Caching? Klarstellung: Dynamische Programmierung und Caching verwenden Ergebnisse für *dieselbe Eingabe* wieder. Der HDA erlaubt es, aus einem Ergebnis *verschiedene Ausgangspunkte* abzuleiten – dies ist ein wesentlicher Unterschied. Caching löst das Neuberechnen desselben Problems; der HDA löst das Entfalten neuer Probleme aus einer Ergebnisstruktur.

Missverständnis 2: Sie sagen „stochastisch und doch geordnet“ – ist das nicht widersprüchlich? Klarstellung: Stochastik und Ordnung können koexistieren. Die Verteilung der Bäume in einem Wald ist stochastisch, aber die Gesamtdichte und die Artenstruktur sind geordnet. Zufälligkeit ist kein Chaos, sondern eine Weise, Struktur zu

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

organisieren. Zufälligkeit im HDA ist intrinsisch und konstruktiv und arbeitet mit der Ordnung zusammen.

Missverständnis 3: Ohne Eingabe und Ausgabe, wie werden Ergebnisse verifiziert? Klarstellung: Der HDA lehnt Eingabe und Ausgabe nicht ab; er vertritt, dass Berechnung nicht im linearen Eingabe-Ausgabe-Modus arbeiten muss. Im „strukturellen Manifestations"-Modus wird die „Gültigkeit" durch strukturelle Konsistenz bestimmt, nicht durch eine Eins-zu-eins-Entsprechung zwischen Eingabe und Ausgabe. Dies gibt die Verifikation nicht auf, sondern schlägt einen anderen Verifikationsrahmen vor.

Missverständnis 4: Ist das nicht einfach Komplexitätstheorie oder Chaostheorie? Klarstellung: Komplexitätstheorie und Chaos beschreiben „Systeme, die schwer vorherzusagen sind". Der HDA versucht, „Systeme zu beschreiben, die durch strukturelle Einstiegspunkte verstanden und gelenkt werden können" – nicht die Vorhersage aufzugeben, sondern die Art und Weise der Vorhersage zu ändern. Die Chaostheorie sagt „Vorhersage ist schwierig"; der HDA fragt: „Können wir die Vorhersage überflüssig machen, indem wir den Einstiegspunkt ändern?"

Missverständnis 5: Sie sagen „Metadaten nicht ändern", aber ist eine Bundfalte nicht auch eine Änderung? Klarstellung: Die Bundfalte ist ein temporärer Zustand, keine permanente Änderung der ursprünglichen Parameter. Die Metadaten (Bundweite, Schrittlänge, Schnitt, Stoff) bleiben unverändert. Dies ist der Unterschied zwischen „Zustandsüberlagerung" und „Attributänderung". Die Falte ist reversibel, temporär und ändert keine wesentlichen Attribute.

Missverständnis 6: Verneint der HDA den Wert traditioneller Algorithmen? Klarstellung: Ganz und gar nicht. Traditionelle Algorithmen sind äußerst wichtig in der menschlichen Technologiegeschichte; ohne sie gäbe es keine modernen Computer, Datenbanken, Kommunikationssysteme, KI, technische Simulation oder Supercomputing. Ihr Wert kann nicht verneint werden. Aber den Wert traditioneller Algorithmen anzuerkennen bedeutet nicht, sie als Endpunkt von Algorithmen zu

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

akzeptieren. Traditionelle Algorithmen lösen Effizienzprobleme innerhalb eines gegebenen Berechnungsparadigmas; der HDA stellt das Berechnungsparadigma selbst in Frage. Das eine ist, wie man besser auf der Straße geht; das andere ist, ob es notwendig ist, diese Straße zu gehen.

Teil Zehn: Der HDA in der Geschichte der menschlichen Berechnung

Um den Lesern zu helfen, den HDA besser in der Geschichte der menschlichen Berechnung zu verorten, gebe ich hier einen kurzen makroskopischen Überblick.

Das menschliche Verständnis von „Berechnung“ hat mehrere Stufen durchlaufen. Die erste Stufe war die manuelle Berechnung: Algorithmen existierten als menschliche Schritte, langsam und fehleranfällig, aber durch diesen Prozess verstanden die Menschen die Grundregeln der Berechnung. Die zweite Stufe war die mechanische Berechnung: von Pascals Rechenmaschine bis zu Babbages Analytical Engine, die Berechnung wurde mechanisiert, aber Algorithmen waren immer noch an physikalische Strukturen gebunden. Die dritte Stufe war die elektronische Berechnung und das Turing-Paradigma: die von-Neumann-Architektur verbreitete sich, Algorithmen wurden als Programme codiert, und die Turingmaschine definierte die Grenzen der Berechenbarkeit. Die vierte Stufe war die parallele Berechnung und Superberechnung: massive Recheneinheiten wurden gestapelt, um komplexere Probleme anzugehen, aber die zugrundeliegende Logik blieb eine Erweiterung des Turing-Paradigmas.

Gegenwärtig befindet sich die Menschheit am Eingang der fünften Stufe. Quantencomputing versucht, die Grenzen klassischer Bits zu überwinden; neuromorphes Computing versucht, die Struktur biologischer Nervensysteme nachzuahmen; der HDA versucht, den linearen Rahmen „Eingabe → Schritte → Ausgabe“ selbst zu durchbrechen.

Der HDA und traditionelle Algorithmen stehen nicht in einer Ersetzungs-, sondern in einer hierarchischen Beziehung. Traditionelle Algorithmen lösen das Problem „wie man

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

auf einem gegebenen Pfad effizienter rechnet“. Der HDA fragt: „Kann der Pfad neu definiert oder sogar umgangen werden?“ Wenn wir die Geschichte der menschlichen Berechnung als einen Prozess der kontinuierlichen Überschreitung eigener Grenzen betrachten, dann ist der HDA ein neuer Knoten in diesem Prozess. Er löst keine Probleme innerhalb des alten Rahmens, sondern schlägt einen neuen Rahmen vor.

Dieses Urteil bedeutet nicht, dass der HDA ausgereift oder vollständig ist. Ganz im Gegenteil: Als neues Konzept befinden sich seine Definitionen, Grenzen, Verifikationsmethoden und Anwendungsszenarien noch im Entstehen. Der Zweck dieses Papiers ist es, den Ursprungspunkt dieses Konzepts zu verankern und eine stabile theoretische Grundlage für die weitere Entwicklung zu liefern.

Teil Elf: Schlussfolgerung – Dies ist keine Optimierung, sondern eine Neudefinition

Der Unterschied zwischen dem HDA und traditionellen Algorithmen ist kein Unterschied von „besser“ vs. „schlechter“, kein Unterschied von „schneller“ vs. „langsamer“, sondern ein grundlegender Unterschied auf Paradigmaebene.

Traditionelle Algorithmen streben nach vorhersehbarer Kontrolle. „Gib mir die Eingabe, und ich werde wissen, welche Ausgabe du erhältst, weil ich jeden Schritt berechnet habe.“ Es ist das Paradigma des Ingenieurs: entwerfen, bauen, testen, verifizieren.

Der HDA beschreibt verständliche Emergenz. „Ich kann nicht jeden Zwischenzustand genau vorhersagen, aber ich weiß, dass das Gesamtmuster sich auf geordnete Weise manifestieren wird. Jeder Datenpunkt ist lebendig, hat eine ‚Perspektive‘.“ Er ähnelt eher dem Paradigma des Ökologen oder Komplexitätstheoretikers: beobachten, verstehen, lenken, emergente Ordnung nutzen.

Traditionelle Algorithmen korrigieren Ergebnisse schrittweise entlang eines gegebenen Pfades. Der HDA macht das gewünschte Ergebnis sofort erreichbar, indem er den strukturellen Einstiegspunkt ändert, wodurch der gesamte Pfad umgangen wird.

Traditionelle Algorithmen „verarbeiten Daten für verschiedene Ergebnisse“. Der HDA „verwendet dieselben Daten, um das Potenzial für mehrere Ergebnisse zu tragen“.

Traditionelle Algorithmen streben danach, „es beim ersten Mal richtig zu machen“. Der HDA strebt danach, „es beim ersten Mal richtig zu machen und es dann nie wieder berechnen zu müssen“.

Dies ist keine Optimierung von Algorithmen; es ist ein Hinterfragen der Prämisse, ob ein Algorithmus notwendigerweise existieren muss.

Daher ist der HDA keine Algorithmusoptimierung, sondern eine Algorithmusneudefinition. Es geht nicht darum, auf der traditionellen Algorithmusspur schneller zu laufen; es geht darum zu fragen, ob es eine andere Spur außerhalb der traditionellen Spur gibt, oder ob eine Spur überhaupt notwendig ist. Es zielt nicht darauf ab zu beweisen, dass traditionelle Algorithmen nutzlos sind, sondern darauf hinzuweisen, dass traditionelle Algorithmen nur eine niedrigdimensionale Form von Algorithmus sind. Es fragt nicht „wie man ein Ergebnis in kürzerer Zeit erhält“, sondern „kann das Ergebnis direkt durch Struktur etabliert werden?“ Es fragt nicht „wie man mehr Daten verarbeitet“, sondern „können dieselben Daten das Potenzial für mehr Ergebnisse tragen?“

Innerhalb der vorherrschenden Berechnungsparadigmen könnte der HDA als unberechenbar, nicht überprüfbar oder schwer in die Grenzen der bestehenden Berechnungstheorie einfügbar angesehen werden. Aber genau dies stellt seinen Paradigmaunterschied dar. Die scheinbare Instabilität eines neuen Konzepts innerhalb eines alten Paradigmas bedeutet nicht notwendigerweise, dass es bedeutungslos ist; es könnte auch bedeuten, dass das alte Paradigma es nicht vollständig aufnehmen kann. Gegenwärtig ist der HDA in erster Linie ein struktureller Vorschlag, eine originäre Definition einer neuen Berechnungsansicht, kein ausgereiftes System mit geschlossener Ingenieursschleife. Er wird spätere Ergänzungen, Entfaltungen, Validierungen und Anwendungen benötigen, aber sein konzeptioneller Ursprung muss zuerst klar artikuliert werden.

Letztendlich zielt der HDA auf eine neue Sichtweise der Berechnung ab: Ein Algorithmus ist nicht notwendigerweise nur Mathematik, nur Schritte, nur ein Programm, nur ein Verarbeitungsprozess zwischen Eingabe und Ausgabe. Ein Algorithmus kann auch eine Organisation mehrdimensionaler Beziehungen sein, ein strukturelles Netzwerk von Daten, Einstiegspunkten, Beziehungen, Zuständen und Ergebnissen. Er kann Ordnung und Zufälligkeit umfassen; er kann von der Ursache zur Wirkung und von der Wirkung zur Ursache gehen; er kann Metadaten unverändert lassen und sich doch an mehrere Zustände anpassen; er kann ein Ergebnis zu einem neuen Ausgangspunkt machen und mehrere Ausgangspunkte in einem einzigen Ergebnis zusammenfließen lassen.

Ein wirklich fortgeschrittener Algorithmus ist nicht notwendigerweise derjenige, der komplexere Berechnungen durchführt, sondern derjenige, der Berechnungen reduzieren, die Struktur lebendig machen, Ergebnisse zu Einstiegspunkten machen und kausale zyklische, mehrdimensionale Organisationsmuster bilden kann.

Dies ist die Kernbedeutung meines Vorschlags des „hyperdimensionalen Algorithmus“. Er ist kein bloßes Neologismus, sondern ein neues Konzept, ein neues Paradigma, eine neue Berechnungsstruktur, die bereits in mehreren realen Systemen etabliert ist. Sein Schwerpunkt liegt nicht auf schnellerer wiederholter Berechnung, sondern auf der Reduzierung redundanter Berechnung; nicht auf der Anhäufung von Rechenleistung, sondern auf der Rekonstruktion von Einstiegspunkten; nicht auf der kontinuierlichen Änderung von Daten, sondern auf der Beibehaltung von Metadaten und der Änderung von Beziehungen; nicht darauf, Ergebnisse als Endpunkte zu belassen, sondern darauf, Ergebnisse zu neuen Ausgangspunkten zu machen. Traditionelle Algorithmen suchen Ergebnisse entlang eines Pfades; der HDA aktiviert Ergebnisse innerhalb einer Struktur. Traditionelle Algorithmen streben danach, Berechnung abzuschließen; der HDA fragt, ob Berechnung in ihrer traditionellen Form existieren muss. Dies ist der grundlegendste Unterschied zwischen ihm und der aktuellen Definition von Algorithmus.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Anhang: Grenzen und offene Fragen dieses Papiers

Dieses Papier präsentiert einen vorläufigen Rahmen für den HDA, keine vollständige formale Definition. Mehrere Fragen bleiben weiter zu erforschen; sie stellen keine Verneinung der hier angebotenen Definition dar, sondern repräsentieren die nächste Forschungsrichtung.

Erstens, Konvergenzbedingungen. Was ist das „Vervollständigungs"-Merkmal für den HDA? Wie beurteilt man, ob ein Ergebnis „gültig" ist? In traditionellen Algorithmen wird die Antwort durch die Eingabe-Ausgabe-Entsprechung definiert. Im HDA muss die Antwort möglicherweise durch strukturelle Konsistenz definiert werden. Diese Definition ist noch nicht abgeschlossen.

Zweitens, Ressourcendarstellung. Wie können Zustände mehrdimensionaler Überlagerung in endlichen Ressourcen dargestellt werden? Sind traditionelle Ressourcenkennzahlen wie Zeit, Raum und Rechenleistung noch anwendbar? Wenn ja, wie sollten sie neu definiert werden? Wenn nicht, welche Kennzahlen ersetzen sie? Diese Fragen warten auf Entwicklung.

Drittens, Gewährleistung der Ordnung. Welche Regeln gewährleisten die „Ordnung" innerhalb des „Stochastischen und doch Geordneten"? Wo ist die Grenze zwischen Stochastik und Ordnung? Unter welchen Umständen zerstört Zufälligkeit die Ordnung? Unter welchen Umständen unterdrückt Ordnung die Zufälligkeit? Diese Fragen müssen weiter untersucht werden.

Viertens, Auswahl bei der Rückwärtsinferenz. Bei der Ableitung mehrerer Ausgangspunkte aus einem Ergebnis, wie werden die verschiedenen Ausgangspunkte ausgewählt oder bewertet? Gibt es ein Maß für „Rückwärtsinferenzeffizienz"? Wenn ja, wie verhält es sich zur Effizienz der Vorwärtsberechnung?

Fünftens, Schnittstelle zu traditionellen Systemen. Wie interagiert der HDA mit dem bestehenden Turingmaschinensystem? Als Ersatz, Ergänzung oder Schichtenansatz? In

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

der praktischen Technik, wie sollte die Grenze zwischen HDA und traditionellen Algorithmen gezogen werden? Gibt es hybride Modelle?

Sechstens, Verifikationsrahmen. Wie werden die Ergebnisse des HDA verifiziert? Wenn Ergebnisse nicht durch lineare Schritte erhalten werden, wie werden Reproduzierbarkeit und Prüfbarkeit etabliert? Erfordert dies eine völlig andere Verifikationsphilosophie?

Diese Fragen sind keine Mängel dieses Papiers, sondern unvermeidliche Merkmale eines „Ankerdokuments". Das Auftauchen eines neuen Paradigmas wird von offenen Fragen begleitet. Dieses Papier ist ein Anfang, kein Abschluss.

Bibliographische Hinweise

Der hier vorgeschlagene „hyperdimensionale Algorithmus" ist keine direkte Erweiterung einer bestehenden akademischen Denkschule. Die folgenden Bereiche bieten jedoch einen Dialogkontext für dieses Papier, der nur zur Orientierung des Lesers dient, nicht als traditionelle akademische Zitate.

Turingmaschinen und Berechenbarkeitstheorie dienen hier als Bezugspunkt für den Vergleich mit traditionellen Algorithmen. **Inverse Probleme und Bayessche Inferenz** stellen bestehende Versuche dar, „die Ursache aus der Wirkung abzuleiten", und obwohl meine „Wirkung-zu-Ursache" verwandt ist, unterscheidet sich ihre Richtung. **Emergenztheorie und komplexe Systeme** beschreiben die „Manifestation von Ergebnissen aus der Struktur", und der HDA versucht, Emergenz „verständlich" und „lenkbar" zu machen. **Wissensgraphen und Graphdatenbanken** sind bestehende Praktiken, bei denen Daten „an mehreren Knoten zusammenlaufen", und der HDA fügt dieser Basis die Dimension der „variablen Einstiegspunkte" hinzu. **Nichtklassische Berechnungsparadigmen**, einschließlich Quanten-, Analog- und neuromorphem Computing, brechen klassische Bits oder Architekturen, während der HDA sich mehr auf das Durchbrechen des linearen Rahmens „Eingabe → Schritte → Ausgabe" selbst konzentriert.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Die Beziehung dieses Papiers zu den oben genannten Bereichen ist eine des Dialogs und der Transzendenz, nicht der Vererbung oder Widerlegung. Das Ziel dieses Papiers ist es nicht, inkrementelle Verbesserungen innerhalb dieser Bereiche vorzunehmen, sondern eine neue Ebene der Fragestellung über ihnen zu erheben.

Related Articles

[Communication] I Have No Algorithm, Yet I Surpass Algorithms!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect–Cause Theory
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Communication] Rejecting Algorithmic Manipulation
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

[Предельный алгоритм] Гипермерный алгоритм

Переопределение самого «вычисления»

Автор: Джеффи Чао Хуэй Ву

Аннотация

В данной статье представлена новая концепция «гипермерного алгоритма» (ГМА) и предлагается структурный пересмотр границ традиционных определений алгоритма. Традиционные алгоритмы обычно строятся на линейной основе ввода, шагов, правил и вывода, подчеркивая конечность, детерминированность, эффективность, осуществимость и четкие границы между вводом и выводом. Хотя современные суперкомпьютеры обладают огромной вычислительной мощью, их базовая логика в основном заключается в выполнении устоявшихся алгоритмических парадигм в большем масштабе. В данной статье утверждается, что действительно заслуживающее внимания «супервычисление» — это не просто наращивание вычислительной мощности, а фундаментальное изменение самой парадигмы вычислений. ГМА — это ни ускоренная версия традиционных алгоритмов, ни расширение математических алгоритмов; это многомерный, суперпозиционный, стохастический, но упорядоченный структурный взгляд на вычисления. В этой структуре данные больше не являются пассивным вводом или конечным выводом, а обладают атрибутом узла, который одновременно является отправной точкой для множества результатов и результатом множества отправных точек. Результаты перестают быть конечными точками и могут становиться новыми точками входа. Метаданные не нужно многократно изменять; посредством структурных точек входа, изменения отношений и условной активации они могут соответствовать различным состояниям и результатам. Используя философию «Следствия-к-Причине» и повседневный пример «складывания пояса брюк», эта статья показывает, как ГМА может делать целевые результаты непосредственно достижимыми путем изменения структурных точек входа, сокращая тем самым избыточные вычисления и даже обходя исходные вычислительные пути. Данная статья 旨在 установить предварительное определение, структурные границы и концептуальные основы для ГМА, предоставив исходный теоретический узел для будущих исследований гипермерных вычислений, предельных алгоритмов и новой научной системы. В соответствии с критерием «существование и обоснованность составляют новую парадигму», ГМА как новая структурная

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

парадигма вычислений, уже систематически проверенная в нескольких областях, является установленной в настоящее время.

Ключевые слова: Гипермерный алгоритм; Предельный алгоритм; Гипермерные вычисления; Следствие-к-Причине; Метаданные; Парадигма вычислений; Структурная точка входа; Нелинейная причинность; Новая наука

Введение: Зачем заново обсуждать «алгоритм»

Я предлагаю «гипермерный алгоритм» (ГМА) не для того, чтобы дать существующим алгоритмам более громкое имя, и не для того, чтобы замаскировать традиционные алгоритмы под новые концепции. Напротив, я ставлю более фундаментальный вопрос: не ограничено ли современное человеческое понимание «алгоритма» слишком узкими рамками?

Сегодня, когда говорят об алгоритмах, обычно думают о математических формулах, программном коде, логических шагах, вводе и выводе, обучении моделей, обработке данных, оптимизации путей, ранжировании поиска, автоматических рекомендациях и логических выводах в искусственном интеллекте. Все это, несомненно, важные формы алгоритмов, которые поддерживают современные компьютеры, Интернет, базы данных, искусственный интеллект, суперкомпьютерные системы, промышленные системы и информационное общество. Однако, если понимать алгоритм лишь как «упорядоченный набор шагов», как процесс, который «начинается с ввода, подвергается обработке по правилам и выдает результат», то такое понимание само по себе ограничивает вычисления низкоразмерными рамками.

Мы живем в эпоху, управляемую алгоритмами. От поисковых систем до рекомендательных систем, от прогноза погоды до ИИ — границы алгоритмов являются границами человеческого интеллекта. Но один вопрос редко задается: Должны ли алгоритмы быть только такими? Почему вычисления должны следовать линейной модели «ввод → шаги → вывод»? Почему одну и ту же задачу нужно каждый раз пересчитывать с нуля? Эта статья пытается бросить вызов этим неявным предположениям и предложить совершенно иную парадигму вычислений — гипермерный алгоритм. В соответствии с критерием «существование и обоснованность составляют новую парадигму», ГМА как новая структурная парадигма вычислений, уже систематически проверенная в нескольких областях, является установленной в настоящее время.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

Необходимо особо пояснить, что предлагаемый здесь «гипермерный алгоритм» полностью отличен от «гипермерных вычислений» (Hyperdimensional Computing, HDC), развивающихся в академических кругах почти три десятилетия. HDC использует случайные двоичные векторы очень высокой размерности для кодирования и вычислений, стремясь к более эффективному представлению и вычислениям, но остается в рамках линейной парадигмы «ввод → обработка → вывод». ГМА же принципиально иной: он ставит вопрос о том, можно ли достичь результата непосредственно через структуру, можно ли обойти сам вычислительный путь и можно ли оставить метаданные неизменными для поддержки множества результатов. Названия звучат похоже, но их смыслы противоположны. ГМА — это не улучшение алгоритмов, а переосмысление предпосылки о том, «должен ли алгоритм обязательно существовать».

Часть первая: Стандартное определение «алгоритма» сегодня

В ведущих областях компьютерных наук, математики и техники существует давно принятое базовое определение алгоритма: алгоритм — это хорошо определенный вычислительный процесс, состоящий из конечного числа шагов, который принимает один или несколько входов, преобразует их с помощью детерминированных правил и выдает один или несколько выходов за конечное время. Это понимание — не чье-то личное мнение, а фундаментальный консенсус, сформировавшийся за долгую историю современной вычислительной цивилизации. Он лежит в основе логики программного обеспечения, аппаратного обеспечения, баз данных, сетевых систем, моделей ИИ и центров суперкомпьютерных вычислений. Как бы сложны ни были алгоритмы, их ядро по-прежнему вращается вокруг ввода, правил, шагов и вывода.

Это стандартное понимание алгоритма можно разложить на несколько ключевых характеристик.

Первое, конечность. Алгоритм должен завершаться после конечного числа шагов; он не может зацикливаться бесконечно или работать вечно. Процесс, который никогда не останавливается, даже если он формально описуем, вряд ли можно считать действительным алгоритмом. Все программы научных вычислений и потоки обработки данных имеют явные условия завершения.

Второе, детерминированность. При одинаковых входных данных и в одинаковых условиях должен выдаваться одинаковый результат. Даже когда некоторые

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

алгоритмы вводят случайность, она обычно является контролируемой, воспроизводимой или статистически интерпретируемой, а не полностью непостижимым внутренним хаосом. Результаты численных симуляций должны быть воспроизводимы — это базовое требование научных вычислений.

Третье, четкие границы между вводом и выводом. Ввод — в начале, обработка — в середине, вывод — в конце. Начальная и конечная точки имеют четкие структурные границы. Вы даете алгоритму данные, он их обрабатывает и выдает вам результат — эта граница ясна, и последовательность фиксирована.

Четвертое, эффективность. Каждый шаг алгоритма должен быть фактически выполнимой операцией, а не скачком, который нельзя выполнить, описать или проверить. Каждый шаг должен быть базовой операцией, которую человек может выполнить с помощью бумаги и карандаша или компьютер может реально выполнить (сложение, вычитание, логический переход, чтение/запись данных и т.д.).

Пятое, осуществимость. Алгоритм, теоретически корректный, но требующий миллиардов лет для выполнения, с инженерной точки зрения не считается осуществимым алгоритмом.

Вся эта совокупность представлений об алгоритме в конечном итоге восходит к модели машины Тьюринга. Как центральная абстракция современной теории вычислений, машина Тьюринга определяет фундаментальную границу «вычислимости». Как бы ни были мощны сегодняшние компьютеры, как бы ни были совершенны их чипы, как бы ни были огромны их суперкомпьютерные центры, их базовая логика не вышла за пределы этого пути: ввод, правила, шаги, вывод. Современные суперкомпьютеры лишь ускоряют этот процесс за счет большего масштаба оборудования, более высокой параллельности и более сложного системного планирования. То есть «супервычисления» в традиционном смысле остаются в первую очередь расширением вычислительной мощности, а не обязательно фундаментальным изменением парадигмы вычислений. Вычислительная мощность может увеличиться в тысячу или десять тысяч раз, но если базовая логика остается «ввод, шаги, вывод», то она все еще находится в рамках традиционной алгоритмической парадигмы.

Часть вторая: Определение «гипермерного алгоритма»

Гипермерный алгоритм, который я предлагаю, возникает именно из этого контекста. Он не является улучшенной версией традиционных алгоритмов, не

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

более быстрым алгоритмом, не более сложной математической формулой и не более совершенной техникой программирования. Это совершенно иное структурное понимание.

Сначала необходимо объяснить значение слова «гипермерный». «Гипермерный» здесь имеет два уровня. Во-первых, он не ограничивается одномерной линейной последовательностью шагов, а позволяет нескольким измерениям разворачиваться одновременно. Во-вторых, он пытается преодолеть традиционные определительные границы «вычисления» — алгоритм больше не обязательно должен быть математическим, упорядоченным или детерминированным. Традиционный алгоритм подобен движению по дороге с односторонним движением: вы должны начать с точки A и пройти через B, C, D, чтобы достичь Z. ГМА не считает, что вычисление должно быть односторонним движением. Он постулирует, что вычисление может быть сетью, полем, многомерной структурой, где любой узел может быть точкой входа и любой узел может быть выходом.

В моей новой научной системе алгоритм не обязательно является математическим алгоритмом, не обязательно является единым упорядоченным вычислением, не обязательно привязан к фиксированным шагам и не обязательно строго следует линейной модели «ввод, обработка, вывод». ГМА — это многомерная, суперпозиционная, стохастическая, но упорядоченная структура. В этой структуре каждая точка данных не является ни изолированным входом, ни фиксированным выходом, а обладает одновременно множеством ролей: она может быть как отправной точкой для множества результатов, так и результатом совместного действия множества отправных точек.

Другими словами, традиционные алгоритмы помещают данные в поток; ГМА помещает данные в структуру. В традиционных алгоритмах данные обычно классифицируются как входные, промежуточные и выходные, каждый с четкой позицией и ролью. Вход — это вход, результат — это результат, промежуточный процесс — это промежуточный процесс. Однако в ГМА одна точка данных не имеет единственной идентичности. Она может быть результатом в одном направлении и отправной точкой в другом; она может быть вызываемым объектом в одной структуре и точкой входа, запускающей новые результаты, в другой структуре. Данные больше не являются пассивным материалом, а многомерным реляционным узлом.

Это точно соответствует ядру ГМА: каждая точка данных является отправной точкой для множества результатов и результатом множества отправных точек. Традиционные алгоритмы используют «упорядоченные шаги» для получения

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

единственного результата; ГМА ближе к многомерной структуре состояний, где результаты не «вычисляются», а естественным образом «проявляются» внутри структуры, в суперпозиции стохастического и упорядоченного. В этой системе данные являются одновременно отправной точкой и компонентом результата.

Чтобы яснее представить ключевые особенности ГМА, я разбиваю их на следующие пять аспектов.

1. Нематематическая природа. Ведущие алгоритмы по своей сути математические — полностью описуемые с помощью символической логики и математических формул. ГМА не обязательно должен быть математическим. Он может основываться на физических принципах, геометрических отношениях, новых парадигмах теории информации или даже принципах сознания. Вычисление может выполняться не через «математические операции», а может естественным образом возникать из геометрических отношений в высокомерном пространстве. Например, форма мыльного пузыря определяется принципом минимизации поверхностного натяжения. Это не «математический алгоритм» вычисляет — это сама физика «вычисляет». ГМА пытается понять этот тип «вычисления», а не имитировать его с помощью математических формул.

2. Многомерность и суперпозиция. Традиционные алгоритмы выполняют упорядоченные шаги в одном измерении, с одним состоянием на шаг. ГМА позволяет нескольким измерениям сосуществовать и нескольким состояниям суперпозироваться одновременно. Алгоритм не следует одному пути, а разворачивается в многомерном поле состояний. Результаты не «вычисляются», а «проявляются» из этого поля. Например, когда снежинка формируется в воздухе, нет «алгоритма», указывающего каждой молекуле воды, куда идти. Расположение молекул является результатом взаимодействия нескольких измерений — температуры, влажности, воздушного потока. Форма снежинки «проявляется», а не «вычисляется».

3. Стохастичность и упорядоченность. Случайность в традиционных алгоритмах является инструментом, внешней, контролируемой помехой; сам алгоритм детерминирован. Случайность в ГМА является внутренней и структурной. Стохастичность и упорядоченность сосуществуют и сотрудничают, вместе образуя сущность алгоритма. Это не «алгоритм со случайностью», а «случайность сама по себе является органическим компонентом алгоритма». Например, экологическая структура леса — распределение деревьев выглядит случайным, но в целом демонстрирует упорядоченную плотность и видовые отношения. Эта «случайность» не является ошибкой, а предпосылкой для нормального функционирования экологической структуры.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

4. Множественные роли данных. В традиционных алгоритмах роль данных уникальна — вход есть вход, выход есть выход, промежуточный результат есть промежуточный результат. В ГМА каждая точка данных является одновременно отправной точкой для множества результатов и результатом множества отправных точек. Узел данных может разворачиваться вниз по течению в несколько путей результатов, и к нему могут сходиться несколько причин вверх по течению. Данные больше не являются точкой в линейном потоке, а узлом пересечения в сети. Например, рассмотрим рост человека. В традиционном алгоритме рост — это «вход» — вы вводите его, чтобы получить такие выходы, как прогноз веса или размер одежды. Но рост также является «результатом» — определяемым несколькими «отправными точками», такими как генетика, питание и физические упражнения. В ГМА точка данных «рост» является одновременно отправной точкой и результатом, а не бинарным выбором.

5. Метаданные неизменны, отношения переменны. Сталкиваясь с различными проблемами, традиционные алгоритмы либо изменяют сами данные, либо пересчитывают все заново. ГМА не изменяет метаданные — основные атрибуты данных остаются неизменными. Изменяются точки входа, интерпретация и отношения между данными. Одна и та же структура метаданных, активируемая через различные точки входа, может давать совершенно разные результаты. Это означает: вычислить один раз, использовать бесконечно много раз. Например, рассмотрим один и тот же текст. Вы можете читать его как поэзию, расшифровывать как шифр или анализировать как исторический документ. Сам текст не изменился — вы изменили только «точку входа». ГМА стремится к этой способности: «одни и те же данные, множественные точки входа, множественные результаты».

С точки зрения структуры данных, вместо изменения самих метаданных, ГМА использует различные точки входа и условия, чтобы заставить ту же самую структуру соответствовать множеству отправных точек и результатов. Данные больше не обрабатываются многократно, а активируются через различные точки входа при сохранении структурной целостности, представляя тем самым различные результаты. Традиция — это «обработка данных для разных результатов»; ГМА — это «использование одних и тех же данных для переноса потенциала множества результатов».

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Часть третья: Фундаментальные различия между ГМА и традиционными алгоритмами

Основываясь на приведенных выше определениях, ГМА и традиционные алгоритмы демонстрируют фундаментальные различия в нескольких ключевых измерениях, подробно описанных ниже.

О сущностной природе: Традиционные алгоритмы являются математическими и формальными, полностью описуемыми с помощью символической логики и математических формул; они по существу являются математическими объектами. ГМА не обязательно должен быть математическим; он может основываться на физических, геометрических, информационных или даже сознательных принципах; он не является подмножеством математики, а более широкой категорией. Это различие можно резюмировать так: от **«математического объекта»** к **«универсальному закону»**.

О временном порядке: Традиционные алгоритмы следуют единому, упорядоченному, линейному временному порядку — шаг A к B к C — строго последовательно или делимы на параллельные, но упорядоченные подшаги, с необратимой стрелой времени. ГМА не имеет фиксированной последовательности шагов; он многомерен, суперпозиционен, стохастичен, но упорядочен. Результаты могут не зависеть от «порядка» выполнения, потому что традиционное понятие «порядка» может не существовать. Это различие можно резюмировать так: от **«линейного времени»** к **«высокомерной одновременности»**.

О причинности: Традиционные алгоритмы подчиняются детерминированной причинно-следственной цепи. При одинаковых входных данных и начальных состояниях выход всегда уникален. Причина предшествует следствию, необратимая односторонняя стрела. ГМА подчиняется суперпозиционной причинности; каждая точка данных является отправной точкой для множества результатов и результатом множества отправных точек. Это отражает мою философию **«Следствия-к-Причине»** — возможно иметь сначала следствие, а потом причину. Результат не является конечной точкой; он может стать новой отправной точкой. Это различие можно резюмировать так: от **«одной причины, одного следствия»** к **«полностью взаимосвязанной причинно-следственной сети»**.

О структуре данных: Традиционные алгоритмы используют одностороннюю структуру потока от ввода к обработке и затем к выводу. Роли данных уникальны с

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

четкими границами; входные данные остаются входными, выходные данные остаются выходными. В ГМА роли данных множественны; одни и те же данные являются одновременно результатом и отправной точкой. Нет абсолютных начальных или конечных точек, только узлы в сети. Это различие можно резюмировать так: от **«однонаправленного потока»** к **«рекурсивной симбиозе»**.

О режиме вычислений: Традиционные алгоритмы пересчитывают все с нуля каждый раз, когда возникает проблема. Даже если аналогичная проблема была решена тысячу раз, тысяча первый раз все равно требует всего процесса — это вычисление без состояния. ГМА, если соответствующий результат существует, не нужно пересчитывать. Он может вывести множество отправных точек из результата и развернуть причинные пути в обратном направлении. Вычисление является осознанным, имеет память и является повторно используемым. Это различие можно резюмировать так: от **«пересчета каждый раз»** к **«однократному повторному использованию»**.

О роли случайности: Случайность в традиционных алгоритмах является псевдослучайной или внешне вводимой; случайные числа — это всего лишь инструменты для выборки, аппроксимации или оптимизации; сам алгоритм детерминирован. Случайность в ГМА является внутренней и конструктивной; это органический компонент, сосуществующий и сотрудничающий с порядком. Это не «случайность внутри алгоритма», а **«случайность является одной из причин, почему алгоритм может работать»**. Это различие можно резюмировать так: от **«инструментальной случайности»** к **«конструктивной случайности»**.

О понимании ресурсов: Традиционные алгоритмы стремятся вычислять быстрее и точнее при заданных ресурсах; ресурсы являются ограничениями, и цель алгоритма — «завершить вычисление в рамках ограничений». ГМА стремится сделать само вычисление ненужным через структурный дизайн. Речь идет не о «вычислении быстрее», а об **«обходе вычисления»**. Ресурсы предназначены не для «потребления», а для **«проектирования точек входа»**. Это различие можно резюмировать так: от **«оптимизации при ресурсных ограничениях»** к **«устранению через структурный дизайн»**.

Часть четвертая: Философия Следствия-к-Причине

Основываясь на приведенных выше сравнениях, мне необходимо развить философию **«Следствия-к-Причине»**, одну из ключевых концептуальных основ ГМА.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Традиционное мышление обычно полагает, что причина предшествует следствию; сначала причина, потом следствие; сначала ввод, потом вывод. Проявлением этой концепции в алгоритмах является: вы должны начать с начала и идти шаг за шагом к концу. Даже если вы знаете ответ, вы не можете напрямую «использовать» этот ответ — потому что у вас нет **«пути доказательства»**.

В моей структуре результат не обязательно является только конечной точкой. Как только результат появляется, он может стать новой отправной точкой, продолжая участвовать в генерации новых структур. Результат может **обратно** участвовать в формировании причин. Из результата можно вывести несколько возможных отправных точек. Причина и следствие больше не расположены однонаправленно, а образуют циклические, сетевые, многомерные отношения.

Другими словами, традиционные алгоритмы спрашивают: **«Имея причину, как получить следствие?»** ГМА спрашивает: **«Имея следствие, как вывести или сконструировать причину(ы)?»**

Необходимо особо пояснить, что утверждение **«следствие приходит первым, затем причина»** не отрицает причинность и не утверждает, что «следствия возникают из ничего». Это означает, что в структурах более высокой размерности причина и следствие могут служить точками входа друг для друга и взаимно преобразовываться. Как в следующем примере с брюками, результат «можно носить, и они не волочатся по земле» определяется первым, а затем я работаю в обратном направлении, чтобы найти точку операции «сложить пояс», которая соответствует аспекту «причины» в структуре. Это не означает, что у следствия нет причины, а что следствие становится новым **пунктом доступа** для обнаружения причины.

Фундаментальное предположение в мире традиционных алгоритмов состоит в том, что время однонаправлено; вы должны вычислять шаг за шагом от начального состояния к конечному. Даже если вы знаете ответ, вы не можете напрямую «использовать» его, потому что у вас нет пути доказательства. ГМА бросает вызов этому предположению: если следствие известно, причины могут быть выведены в обратном направлении; одно и то же следствие может соответствовать нескольким причинным путям; вычисление больше не является путем от начала к концу, а **развертыванием всех возможных отправных точек с конца**.

Часть пятая: Повседневный пример – Брюки и складывание пояса

Чтобы более интуитивно понять вышеизложенные абстрактные различия, я использую повседневный пример.

Человек покупает новые брюки с эластичным поясом. Штанины немного длинноваты, волочатся по полу и создают неудобства.

Традиционный подход: заметив, что штанины слишком длинные, подворачивает часть штанины, зашивает иголкой с ниткой, затем примеряет. Если ребенок подрастет и брюки станут короткими, зашитая штанина не может быть опущена, и брюки приходят в негодность. В следующий раз, когда покупаются новые брюки, процесс повторяется. Этот метод кажется естественным, потому что проблема, по-видимому, находится на штанинах, поэтому он воздействует на штанины. Это соответствует традиционному алгоритмическому мышлению: определить проблему, локализовать симптом, выполнить обработку в месте симптома и, наконец, получить результат. Штанины слишком длинные → изменить штанины; данные неверны → изменить данные; путь неподходящий → продолжить исправление вдоль пути.

Мой подход иной. Вместо того чтобы изменять подгиб, я складываю пояс, и брюки становятся сразу пригодными для носки. Это действие очень простое, но его основная структурная логика совершенно иная. Я не режу подгиб, не зашиваю его навсегда, не меняю исходную длину и не повреждаю метаданные (обхват талии, длина шагового шва, крой, ткань). Я просто изменяю пояс, который является **структурной точкой входа**. Регулируя эту глобальную контрольную точку для всего состояния носки, проблема ниже по течению (слишком длинные штанины) исчезает немедленно.

Более важно то, что этот метод является **обратимым, регулируемым и многоразовым**. Это особенно очевидно для растущего ребенка. Если ребенок в настоящее время недостаточно высок, и брюки слегка длинны, я складываю пояс один раз. Когда ребенок позже подрастет, я складываю пояс меньше. Когда ребенок подрастает еще больше, я полностью разворачиваю складку. Исходная структура брюк остается нетронутой, но адаптируется к меняющемуся росту ребенка. Традиционный метод зашивания подгиба может привести к тому, что брюки станут слишком короткими, когда ребенок подрастет; если их обрезать, это необратимо. Мой метод оставляет метаданные неизменными, позволяя **одной и той же структуре адаптироваться к нескольким состояниям**.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي النسخة الصينية العربية.
Chinesische Originalfassung maßgeblich.

Этот пример выявляет несколько важных структурных различий.

Традиционный подход соответствует традиционной алгоритмической логике: проблема в подгибе, поэтому подгиб должен быть изменен, действуя шаг за шагом по пути от «причины к следствию», не пропуская ни одного шага. Решает одну проблему за раз, необратимо, и требует повторения всего процесса, когда та же проблема возникает снова. Мой подход соответствует логике ГМА: цель — избежать волочения. Вместо того чтобы решать поверхностную причину «штанины слишком длинные», я нахожу **глобальную контрольную точку** — пояс. Однократное складывание пояса немедленно укорачивает эффективную длину штанин, и проблема исчезает. Я не изменяю никаких метаданных; складка — это всего лишь временное, обратимое, динамическое состояние.

С точки зрения данных, традиционный подход изменяет метаданные — зашивание штанин навсегда укорачивает их; исходные данные теряются. Мой подход не изменяет никаких метаданных; исходные размеры остаются полностью нетронутыми; я лишь добавляю временное, обратимое, динамическое состояние складки. Складка на поясе не создает новые данные и не уничтожает старые данные; она лишь **перестраивает отношения** между данными — уменьшает эффективный обхват талии и косвенно изменяет эффективную длину штанины.

В этом примере «штанины брюк не волочатся по земле» является желаемым результатом. Традиционный метод исходит из причины «штанины слишком длинные», изменяет подгиб и в конечном итоге достигает результата «не волочатся». Мой метод сначала проясняет результат «не волочатся», затем работает в обратном направлении, чтобы найти структурную точку входа, и обнаруживает, что простое складывание пояса делает результат достижимым. Это практическое воплощение **«Следствия-к-Причине»**. Необязательно идти шаг за шагом от причины к следствию; можно **работать от следствия, чтобы определить структуру**, делая исходный путь ненужным.

Этот пример также отвечает на вопрос о **«метаданных»**. Что такое метаданные? В примере с брюками метаданные — это исходные размеры: обхват талии, длина шагового шва, крой, ткань — **«существенные атрибуты»** брюк. Временные данные — это складка на поясе; она не изменяет ни одного из исходных размеров, а лишь временно складывает секцию. Традиционный подход изменяет метаданные — длина шагового шва навсегда укорачивается; исходные данные теряются. Мой подход не изменяет никаких метаданных — исходные размеры брюк остаются полностью нетронутыми; я лишь добавляю временное, обратимое, динамическое состояние.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Исходный обхват талии как метаданное является одновременно **«отправной точкой для множества результатов»**: он поддерживает множественные состояния носки — нормальная носка, носка с одной складкой, носка с двумя складками. Он также является **«результатом множества отправных точек»**: определяется выбором ткани, методом кроя, дизайнерским замыслом и т.д. Складка на поясе как временное данное не создает новые данные и не уничтожает старые данные; она лишь перестраивает отношения.

Это не просто трюк, а **структурное мышление**. Многие люди, видя слишком длинные штанины, притягиваются к симптому штанин, поэтому все лечение концентрируется на штанинах. Однако с точки зрения общей структуры, длина штанин — это всего лишь проявление ниже по течению; изменение положения пояса может повлиять на то, как брюки сидят. То есть **проблему не обязательно решать в том месте, где она проявляется**. Многие низкоразмерные проблемы имеют свои истинные контрольные точки на более высоком структурном уровне. Традиционные алгоритмы склонны продолжать вычисление вдоль поверхности проблемы; ГМА ищет **структурную точку входа**.

Это также объясняет, почему ГМА не обязательно сложнее, а может быть **проще**. Действительно высокоуровневые структуры не обязательно усложняют проблемы, а делают сложные проблемы простыми в правильной точке входа. Столкнувшись со сложной проблемой, традиционный алгоритм может добавлять шаги, модели, вычислительную мощность, память, время обучения. ГМА ищет структурный узел; как только этот узел правильно активирован, изначально сложный путь может стать ненужным. **«Обход пути»** — это не лень, а переопределение необходимости этого пути.

В этом примере: традиционный подход **«изменяет результат вдоль пути»**, в то время как ГМА **«изменяет точку входа, делая весь путь неэффективным»**. Обычный метод латает на стороне результата; другой метод напрямую изменяет структурную точку входа, так что проблемы, требующие повторного лечения, перестраиваются в начальной точке. Традиционный ведущий алгоритм идет от **«начальной точки вдоль пути к результату»**; в ГМА **«сам результат может стать точкой входа в путь и участвовать в генерации новых структур»**. Традиция — это **«одна структура для одного результата»**; ГМА — это **«одна структура, несущая множественные состояния результатов»**.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Часть шестая: Взгляд на данные, основанный на неизменности метаданных

С точки зрения данных, у ГМА есть очень важный принцип: **не изменять метаданные, делая их применимыми к множеству отправных точек или результатов**.

Метаданные — это исходные атрибуты данных, фундаментальная информация, онтологическая информация данных. Сталкиваясь с различными проблемами, традиционные подходы часто изменяют данные, копируют их, обрабатывают, пересчитывают или строят разные пути для разных результатов. Эти подходы могут решать проблемы, но ценой непрерывной обработки данных, повторения путей и постоянного роста сложности системы.

ГМА, напротив, подчеркивает, что при максимальном сохранении метаданных неизменными, одна и та же структура метаданных может соответствовать множеству отправных точек и множеству результатов путем изменения точек входа, отношений, условий и методов активации. Онтология данных остается статичной; отношения меняются. Фундаментальная структура остается неизменной; представление меняется. Метаданные остаются нетронутыми, но адаптируются к различным состояниям.

Это то, что я называю **«вычислить один раз, использовать бесконечно много раз»**. «Вычислить один раз» здесь — это не просто кэширование результатов; это означает, что однажды установленная структура больше не должна перестраивать полный путь для каждого состояния. Одна и та же структура данных может быть активирована через различные точки входа для получения различных результатов. Это не «обработка данных для разных результатов», а **«использование одних и тех же данных для переноса потенциала множества результатов»**. Это также одно из ключевых отличий ГМА от традиционных алгоритмов. Традиционные алгоритмы обрабатывают данные на пути; ГМА обрабатывает общую структуру данных, точек входа, отношений, состояний и результатов.

В своей минимальной структуре ГМА может быть понят как: система, которая без изменения метаданных заставляет один и тот же узел данных соответствовать множеству путей результатов за счет изменения структурных точек входа. Это определение не стремится немедленно свести ГМА к традиционной математической формуле, а сначала установить его структурные границы. Это не единственная линейная функция, не фиксированная формула и не простой

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

механизм кэширования. Это **реляционная структура**: метаданные остаются стабильными; точки входа, условия, отношения и результаты могут меняться и проявляться в нескольких направлениях. Именно в этой структуре вычисление перестает быть mere выполнением шагов и становится **активацией отношений**.

С точки зрения данных, сущность **«не изменять метаданные, чтобы сделать их применимыми к множеству отправных точек или результатов»** такова: онтология данных остается неизменной; изменения происходят в **«точке входа интерпретации/использования»**. В традиционных структурах изменение проблемы ведет к изменению данных или пути. В ГМА изменение проблемы ведет к изменению **интерпретационной структуры**, а не данных.

Часть седьмая: Переопределение «супервычисления»

В этой системе мне нужно прояснить термин — **«супервычисление»**.

То, что я подразумеваю под «супервычислением», — это не обычное значение «суперкомпьютера»: не больше чипов, не более высокая вычислительная мощность, не большие центры обработки данных. То, что я подразумеваю под «супервычислением», — это **«гипермерный алгоритм»**.

Истинное супервычисление может быть не накоплением вычислительной мощности, а **трансформацией парадигмы вычислений**. Когда система все еще полагается на повторяющиеся вычисления, какой бы мощной она ни была, она остается в ловушке внутри пути. Когда система может структурно сокращать вычисления, обходить повторяющиеся пути и делать результаты новыми точками входа, она начинает приближаться к истинному гипермерному вычислению.

Другими словами, традиционное супервычисление стремится **«использовать больше вычислительной мощности для решения больших проблем»**. ГМА стремится **«использовать лучшую структуру, чтобы проблема больше не требовала такой большой вычислительной мощности»**. Эти два подхода не противоречат друг другу, но их направления различаются.

Если традиционное супервычисление представляет пределы вычислительной мощности, то ГМА представляет **поворот в понимании вычислений**. Истинное «супервычисление» может быть не большим количеством чипов, более крупными центрами обработки данных, более высоким энергопотреблением или более сложными моделями. Истинное супервычисление может быть **обнаружением структурной точки входа**, **устранением пути**, **обратным развертыванием узла**

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

результата или **одновременным обеспечением условий для множества результатов без изменения метаданных**. Чем сильнее вычислительная мощность, если она остается в ловушке низкоразмерных путей, она является мощной только внутри пути. Как только структура будет повышена, даже чрезвычайно простые операции могут дать эффективность более высокого уровня.

Супервычисление — это предел вычислительной мощности; гипермерный алгоритм — это **переопределение предпосылки «должны ли вычисления зависеть от вычислительной мощности»**.

Часть восьмая: Эмпирическая основа – Систематическая валидация в нескольких областях и определение новой парадигмы

ГМА — это не чисто теоретическое построение. Он уже был валидирован в нескольких реальных системах.

Моя **логистическая система** не требует вычислительной мощности суперкомпьютера или облачной поддержки; она выполняет сложное планирование с скромными ресурсами. Результаты многоразовы, структура многократно адаптируема, и нет необходимости пересчитывать все с нуля каждый раз. Она работает без суперкомпьютера или облака, выполняя сложное планирование с низкими ресурсами — это установленный и ценный прорыв в инженерии. Это демонстрирует, что высокая вычислительная мощность — не единственное решение; структурный дизайн может заменить часть вычислительной мощности.

Моя **издательская система** также использует ту же структурную логику, достигая эффективности, превосходящей любую существующую систему. Моя **система генерации веб-страниц «Предел»** также основана на том же ГМА, многократно доказывая в нескольких областях, что та же структура может стабильно существовать. Сотрудники моей компании уже используют эти системы, что показывает, что эта структура может функционировать без моего постоянного личного вмешательства.

Общая черта этих систем: они не полагаются на ведущие пути высокой вычислительной мощности, не следуют фиксированным вычислительным шагам и достигают желаемого результата непосредственно за счет настройки структурных

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

точек входа. Это не разовые успехи или изолированные трюки, а **повторное возникновение одной и той же структурной логики в разных областях**.

Это указывает на то, что ГМА не случаен, а представляет собой **структурную методологию, уже валидированную несколькими системами**. Это не «личная техника», а парадигма вычислений, стабильно установленная в таких областях, как логистика, издательское дело и генерация веб-страниц.

В соответствии с критерием **«существование и обоснованность составляют новую парадигму»**, ГМА как новая структурная парадигма вычислений, уже систематически проверенная в нескольких областях, является установленной в настоящее время. Я не выдвигаю гипотезу; я описываю другую структуру вычислений, уже работающую в нескольких системах. Мне не нужно доказывать, что она понята; я доказал, что она стабильно существует в разных системах. В настоящее время, в рамках моего практического применения, эта структура уже может рассматриваться как новая парадигма.

Что касается критериев для определения ГМА как новой парадигмы: когда структура логически непротиворечива, стабильно существует в нескольких системах и демонстрирует **несводимые различия** с существующими методами, она уже обладает основами новой парадигмы. **Самосогласованность** — это отправная точка; **эмпирическое доказательство** — это основа; **структурное различие** — это ядро парадигмы. Согласно критерию «существование и обоснованность», эта система уже является новой парадигмой. Признает ли это внешний мир, влияет только на распространение, а не на обоснованность. Это уже установленная в реальных системах парадигма структурных вычислений, обоснованность которой не зависит от внешнего признания или консенсуса.

Фундаментальное различие между ГМА и ведущими парадигмами вычислений заключается в следующем: традиционные алгоритмы в основном используют **прямое** вычисление; как только результат получен, он обычно не может обратно участвовать в новых структурах рассуждений. В ГМА результат больше не является конечной точкой, а **многоразовым структурным узлом**. При определенных условиях результаты могут участвовать в новых путях рассуждений, сокращая избыточные вычисления и формируя вычислительные сети с **множеством отправных точек и множеством путей**. В традиционных вычислительных структурах результат обычно является конечной точкой; в структуре ГМА сам результат может стать новой отправной точкой, формируя многопутевую сеть рассуждений. ГМА — это не улучшение алгоритмов; это **переосмысление предпосылки о том, должен ли алгоритм обязательно существовать**.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

Часть девятая: Разъяснение распространенных заблуждений

Основываясь на предварительных беседах с читателями из разных областей, я предвижу следующие шесть заблуждений и заранее разъясняю их, чтобы избежать преждевременного насильственного втискивания концепции в неподходящие рамки.

Заблуждение 1: Разве ГМА — это не просто динамическое программирование или кэширование?
Разъяснение: Динамическое программирование и кэширование повторно используют результаты для *одного и того же входа*. ГМА позволяет выводить *различные отправные точки* из результата — это существенное различие. Кэширование решает проблему пересчета той же задачи; ГМА решает проблему развертывания новых задач из структуры результата.

Заблуждение 2: Вы говорите «стохастический, но упорядоченный» — разве это не противоречие?
Разъяснение: Стохастичность и упорядоченность могут сосуществовать. Распределение деревьев в лесу стохастично, но общая плотность и видовая структура упорядочены. Случайность — это не хаос, а способ организации структуры. Случайность в ГМА является внутренней и конструктивной и работает в сотрудничестве с порядком.

Заблуждение 3: Без ввода и вывода, как проверить результаты?
Разъяснение: ГМА не отвергает ввод и вывод; он утверждает, что вычисления не должны работать в линейном режиме ввода-вывода. В режиме «структурной манифестации» «валидность» определяется **структурной согласованностью**, а не однозначным соответствием между вводом и выводом. Это не отказ от верификации, а предложение иной системы верификации.

Заблуждение 4: Разве это не просто теория сложности или теория хаоса?
Разъяснение: Теория сложности и хаоса описывают «системы, которые трудно предсказать». ГМА пытается описать «системы, которые можно понять и направлять через структурные точки входа» — не отказываясь от предсказания, а **меняя способ предсказания**. Теория хаоса говорит: «Предсказание сложно»; ГМА спрашивает: **«Можем ли мы сделать предсказание ненужным, изменив точку входа?»**

Заблуждение 5: Вы говорите «не изменяйте метаданные», но разве складка на поясе не является изменением?
Разъяснение: Складка на поясе — это временное состояние, а не постоянное

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

изменение исходных параметров. Метаданные (обхват талии, длина шагового шва, крой, ткань) остаются неизменными. Это различие между **«суперпозицией состояний»** и **«модификацией атрибута»**. Складка обратима, временна и не меняет существенных атрибутов.

Заблуждение 6: Отрицает ли ГМА ценность традиционных алгоритмов?
Разъяснение: Абсолютно нет. Традиционные алгоритмы чрезвычайно важны в истории человеческих технологий; без них не было бы современных компьютеров, баз данных, систем связи, ИИ, инженерного моделирования или суперкомпьютеров. Их ценность нельзя отрицать. Но признание ценности традиционных алгоритмов не означает принятие их как конечной точки алгоритмов. Традиционные алгоритмы решают проблемы эффективности внутри данной парадигмы вычислений; ГМА ставит под вопрос **саму парадигму вычислений**. Один вопрос: **как лучше идти по дороге**; другой вопрос: **нужно ли вообще идти по этой дороге**.

Часть десятая: ГМА в истории человеческих вычислений

Чтобы помочь читателям лучше расположить ГМА в истории человеческих вычислений, я даю здесь краткий макроскопический обзор.

Человеческое понимание «вычислений» прошло через несколько этапов.

Первый этап был ручным вычислением: алгоритмы существовали как человеческие шаги, медленные и подверженные ошибкам, но через этот процесс люди поняли основные правила вычислений.

Второй этап был механическим вычислением: от машины Паскаля до аналитической машины Бэббиджа, вычисления были механизированы, но алгоритмы все еще были привязаны к физическим структурам.

Третий этап был электронным вычислением и парадигмой Тьюринга: архитектура фон Неймана стала повсеместной, алгоритмы были закодированы как программы, а машина Тьюринга определила границы вычислимости.

Четвертый этап был параллельным вычислением и супервычислениями: массивные вычислительные блоки объединялись для решения более сложных проблем, но базовая логика оставалась расширением парадигмы Тьюринга.

В настоящее время человечество находится на входе в **пятый этап**. Квантовые вычисления пытаются преодолеть ограничения классических битов;

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

нейроморфные вычисления пытаются имитировать структуру биологических нервных систем; ГМА пытается сломать сам линейный каркас **«ввод → шаги → вывод»**.

ГМА и традиционные алгоритмы находятся не в отношении замены, а в **иерархическом отношении**. Традиционные алгоритмы решают, **«как более эффективно вычислять на данном пути»**. ГМА спрашивает: **«Может ли путь быть переопределен или даже обойден?»** Если рассматривать историю человеческих вычислений как процесс непрерывного преодоления собственных границ, то ГМА — это новый узел в этом процессе. Он не решает проблемы внутри старой структуры, а **предлагает новую структуру**.

Это суждение не означает, что ГМА является зрелым или завершенным. Напротив: как новая концепция, его определения, границы, методы верификации и сценарии применения все еще формируются. Цель этой статьи — именно **закрепить исходную точку этой концепции**, обеспечив стабильную теоретическую основу для последующего развития.

Часть одиннадцатая: Заключение – Это не оптимизация, а переопределение

Различие между ГМА и традиционными алгоритмами — это не различие «лучше» и «хуже», не «быстрее» и «медленнее», а **фундаментальное различие на уровне парадигмы**.

Традиционные алгоритмы стремятся к **предсказуемому контролю**. «Дай мне вход, и я буду знать, какой выход ты получишь, потому что я вычислил каждый шаг». Это **парадигма инженера**: проектировать, строить, тестировать, проверять.

ГМА описывает **понятную эмерджентность**. «Я не могу точно предсказать каждое промежуточное состояние, но я знаю, что общий паттерн проявится упорядоченным образом. Каждая точка данных жива, имеет «перспективу»». Это ближе к **парадигме эколога или теоретика сложных систем**: наблюдать, понимать, направлять, использовать возникающий порядок.

Традиционные алгоритмы **постепенно корректируют результаты** вдоль заданного пути. ГМА делает желаемый результат **немедленно достижимым**, изменяя структурную точку входа, тем самым **обходи весь путь**.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Традиционные алгоритмы **«обрабатывают данные для разных результатов»**. ГМА **«использует одни и те же данные для переноса потенциала множества результатов»**.

Традиционные алгоритмы стремятся **«сделать правильно один раз»**. ГМА стремится **«сделать правильно один раз, а потом никогда больше не вычислять»**.

Это не оптимизация алгоритмов; это **переосмысление предпосылки о том, «должен ли алгоритм существовать»**.

Следовательно, ГМА — это не оптимизация алгоритмов, а **переопределение алгоритмов**. Речь идет не о том, чтобы бежать быстрее по традиционной алгоритмической дорожке; речь идет о том, чтобы спросить, существует ли другой путь за пределами традиционного пути, или даже **нужен ли путь вообще**. Он не стремится доказать, что традиционные алгоритмы бесполезны, а указывает на то, что традиционные алгоритмы — это лишь **низкоразмерная форма** алгоритма. Он не спрашивает **«как получить результат за меньшее время»**, а спрашивает: **«Может ли результат быть непосредственно установлен через структуру?»** Он не спрашивает **«как обработать больше данных»**, а спрашивает: **«Могут ли те же данные нести потенциал для большего количества результатов?»**

В рамках ведущих парадигм вычислений ГМА может рассматриваться как **невычислимый, непроверяемый** или трудно вписывающийся в границы существующей теории вычислений. Но именно это составляет его **парадигмальное отличие**. Кажущаяся нестабильность новой концепции внутри старой парадигмы не обязательно означает, что она бессмысленна; это может также означать, что старая парадигма не может ее полностью вместить. В настоящее время ГМА — это прежде всего **структурное предложение**, **исходное определение нового взгляда на вычисления**, а не зрелая система с замкнутым инженерным циклом. Он потребует последующих дополнений, развертываний, валидаций и приложений, но его **концептуальное происхождение должно быть сначала четко сформулировано**.

В конечном счете, то, на что указывает ГМА, — это **новый взгляд на вычисления**: алгоритм не обязательно является just математикой, just шагами, just программой, just процессом обработки между входом и выходом. Алгоритм также может быть **организацией многомерных отношений**, **структурной сетью данных, точек входа, отношений, состояний и результатов**. Он может включать порядок и случайность; он может идти от причины к следствию и от следствия к причине;

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

он может оставлять метаданные неизменными, адаптируясь к нескольким состояниям; он может сделать один результат новой отправной точкой и позволить нескольким отправным точкам сходиться в одном результате.

Действительно продвинутый алгоритм — это не обязательно тот, который выполняет более сложные вычисления, а тот, который может сократить вычисления, сделать структуру живой, сделать результаты точками входа и сформировать циклические, многомерные паттерны организации причинности.

Это и есть основное значение моего предложения **«гипермерного алгоритма»**. Это не просто неологизм, а **новая концепция, новая парадигма, новая структура вычислений, уже установленная в нескольких реальных системах**. Его акцент — не на более быстрых повторяющихся вычислениях, а на **сокращении избыточных вычислений**; не на накоплении вычислительной мощности, а на **реконструкции точек входа**; не на непрерывном изменении данных, а на **сохранении метаданных и изменении отношений**; не на оставлении результатов в качестве конечных точек, а на **превращении результатов в новые отправные точки**. Традиционные алгоритмы ищут результаты на пути; ГМА **активирует результаты внутри структуры**. Традиционные алгоритмы стремятся завершить вычисление; ГМА спрашивает, **должны ли вычисления существовать в своей традиционной форме**. Это и есть самое фундаментальное различие между ним и текущим определением алгоритма.

Приложение: Границы и открытые вопросы данной статьи

В данной статье представлен предварительный каркас для ГМА, а не полное формальное определение. Следующие вопросы остаются для дальнейшего изучения; они не являются отрицанием предложенного здесь определения, а представляют собой следующее направление исследований.

1. Условия сходимости. Что является маркером «завершения» для ГМА? Как судить, является ли результат «валидным»? В традиционных алгоритмах ответ определяется соответствием ввода и вывода. В ГМА ответ, возможно, должен определяться **структурной согласованностью**. Это определение еще не завершено.

2. Представление ресурсов. Как состояния многомерной суперпозиции могут быть представлены в конечных ресурсах? Применимы ли еще традиционные

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

метрики ресурсов, такие как время, пространство и вычислительная мощность? Если да, как их следует переопределить? Если нет, какие метрики их заменят? Эти вопросы ждут разработки.

3. Гарантия порядка. Какие правила гарантируют «порядок» внутри «стохастического, но упорядоченного»? Где граница между стохастичностью и порядком? При каких обстоятельствах случайность разрушает порядок? При каких обстоятельствах порядок подавляет случайность? Эти вопросы требуют дальнейшего изучения.

4. Отбор при обратном выводе. При выводе нескольких отправных точек из результата, как отбирать или оценивать различные отправные точки? Существует ли мера **«эффективности обратного вывода»**? Если да, как она соотносится с эффективностью прямого вычисления?

5. Интерфейс с традиционными системами. Как ГМА взаимодействует с существующей системой машин Тьюринга? Как замена, дополнение или слоистый подход? В практической инженерии, как должна проводиться граница между ГМА и традиционными алгоритмами? Существуют ли гибридные модели?

6. Система верификации. Как верифицировать результаты ГМА? Если результаты получаются не через линейные шаги, как установить воспроизводимость и проверяемость? Требует ли это совершенно иной философии верификации?

Эти вопросы не являются недостатками данной статьи, а **неизбежными характеристиками «якорного документа»**. Появление любой новой парадигмы сопровождается открытыми вопросами. Эта статья — начало, а не заключение.

Библиографические примечания

Предлагаемый здесь «гипермерный алгоритм» не является прямым продолжением какой-либо существующей академической школы мысли. Однако следующие области обеспечивают контекст для диалога с данной статьей и предлагаются только для ориентации читателя, а не как традиционные академические цитаты.

Машины Тьюринга и теория вычислимости служат здесь ориентиром для сравнения с традиционными алгоритмами.

Обратные задачи и байесовский вывод представляют собой существующие попытки «вывести причину из следствия», и хотя моя философия «Следствия-к-Причине» связана с ними, ее направление иное.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

Теория эмерджентности и сложные системы описывают «проявление результатов из структуры», и ГМА пытается сделать эмерджентность «понятной» и «направляемой».

Графы знаний и графовые базы данных являются существующей практикой, где данные «сходятся в нескольких узлах», и ГМА добавляет к этой основе измерение «переменных точек входа».

Неклассические парадигмы вычислений, включая квантовые, аналоговые и нейроморфные вычисления, преодолевают классические биты или архитектуры, в то время как ГМА больше фокусируется на преодолении самого линейного каркаса «ввод → шаги → вывод».

Отношение данной статьи к вышеуказанным областям — это **диалог и трансценденция**, а не наследование или опровержение. Цель данной статьи — не вносить инкрементальные улучшения внутри этих областей, а **поднять новый уровень вопрошания над ними**.

Related Articles
[Communication] I Have No Algorithm, Yet I Surpass Algorithms!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect–Cause Theory
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Communication] Rejecting Algorithmic Manipulation
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

[극한 알고리즘] 초차원 알고리즘

'계산' 자체의 재정의

저자: 제피 차오 후이 우 (Jeffi Chao Hui Wu)

요약

본 논문은 '초차원 알고리즘'이라는 새로운 개념을 제시하고, 이를 기존 알고리즘 정의의 경계에 대한 구조적 재검토로 제안한다. 기존 알고리즘은 일반적으로 입력, 단계, 규칙, 출력의 선형적 프레임워크 위에 구축되며, 유한성, 결정성, 유효성, 실현 가능성, 명확한 입출력 경계를 강조한다. 현대 슈퍼컴퓨터는 막대한 계산 능력을 보유하고 있지만, 그 기본 논리는 주로 기존 알고리즘 패러다임을 더 큰 규모로 실행하는 데 있다. 본 논문은 진정으로 논의할 가치가 있는 '초계산'은 단순한 계산 능력의 확장이 아니라 계산 패러다임 자체의 변화라고 주장한다. 초차원 알고리즘은 기존 알고리즘의 가속 버전도, 수학적 알고리즘의 확장도 아니다. 그것은 다차원적이고, 중첩적이며, 무작위적이면서도 질서 있는 구조적 계산관이다. 이 구조에서 데이터는 더 이상 수동적인 입력이나 최종 출력이 아니라, 여러 결과의 시작점이자 여러 시작점의 결과인 노드의 속성을 동시에 갖는다. 결과는 더 이상 종착점이 아니라 새로운 진입점이 될 수 있다. 메타데이터는 반복적으로 수정될 필요 없이, 구조적 진입점, 관계 변화, 조건부 활성화를 통해 다양한 상태와 결과에 대응할 수 있다. 본 논문은 '결과-원인' 철학과 '바지 허리 접기'라는 일상적 사례를 통해 초차원 알고리즘이 구조적 진입점을 변경함으로써 목표 결과를 직접 성립시키고, 중복 계산을 줄이며, 기존 계산 경로를 우회할 수 있음을 보여준다. 본 논문은 초차원 알고리즘에 대한 초기 정의, 구조적 경계, 사상적 기초를 확립하고, 향후 초차원 계산, 극한 알고리즘, 새로운 과학 체계를 위한 원시적 이론 노드를 제공하는 것을 목표로 한다. '존재하고 성립하는 것이 곧 새로운 패러다임'이라는 기준에 따라, 초차원 알고리즘은 이미 여러 영역에서 체계적으로 검증된 새로운 계산 구조 패러다임으로서 현재 이미 성립되었다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

핵심어: 초차원 알고리즘; 극한 알고리즘; 초차원 계산; 결과-원인; 메타데이터; 계산 패러다임; 구조적 진입점; 비선형 인과; 새로운 과학

서론: 왜 '알고리즘'을 다시 논의하는가

내가 '초차원 알고리즘'을 제안하는 것은 기존 알고리즘에 더 큰 이름을 붙이기 위해서나, 전통적 알고리즘을 새로운 개념으로 포장하기 위해서가 아니다. 오히려 나는 더 근본적인 질문을 제기한다. 즉, 인간의 현재 '알고리즘'에 대한 이해가 너무 좁은 틀에 갇혀 있는 것은 아닌가?

오늘날 사람들이 알고리즘을 말할 때, 일반적으로 수학 공식, 프로그램 코드, 논리적 단계, 입출력, 모델 훈련, 데이터 처리, 경로 최적화, 검색 순위, 자동 추천, 인공지능 추론 등을 떠올린다. 이들은 물론 현대 컴퓨터, 인터넷, 데이터베이스, 인공지능, 초계산, 산업 시스템, 정보 사회를 지탱하는 중요한 알고리즘 형태이다. 그러나 알고리즘을 단지 '정렬된 단계들의 집합', 즉 '입력으로 시작하여 규칙 기반 처리를 거쳐 출력을 얻는' 과정으로만 이해한다면, 이러한 이해 자체가 계산을 저차원의 틀 안에 가두는 것이다.

우리는 알고리즘이 지배하는 시대에 살고 있다. 검색 엔진에서 추천 시스템, 일기 예보에서 인공지능에 이르기까지, 알고리즘의 경계는 인간 지능의 경계이다. 그러나 거의 묻지 않는 질문이 있다. 알고리즘은 반드시 이래야 하는가? 왜 계산은 '입력 → 단계 → 출력'이라는 선형 모델을 따라야 하는가? 왜 같은 문제를 매번 처음부터 다시 계산해야 하는가? 본 논문은 이러한 암묵적 가정에 도전하고, 완전히 다른 계산 패러다임인 초차원 알고리즘을 제안한다. '존재하고 성립하는 것이 곧 새로운 패러다임'이라는 기준에 따라, 초차원 알고리즘은 이미 여러 영역에서 체계적으로 검증된 새로운 계산 구조 패러다임으로서 현재 이미 성립되었다.

특히, 본 논문에서 제안하는 '초차원 알고리즘'은 학계에서 거의 30 년간 발전해 온 '초차원 계산'(HDC)과 완전히 다른 개념임을 밝혀둔다. HDC 는 매우 높은 차원의 무작위 이진 벡터를 사용하여 인코딩 및 연산을 수행함으로써 더 효율적인 표현과 계산을 추구하지만, 여전히 '입력 → 처리 → 출력'이라는 선형 패러다임 내에 있다. 반면 초차원 알고리즘은 근본적으로 다르다. 그것은 결과가 구조를 통해 직접 성립될

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

수 있는지, 계산 경로 자체를 우회할 수 있는지, 메타데이터를 수정하지 않고 여러 결과에 적응할 수 있는지를 묻는다. 이름은 유사하지만, 그 의미는 반대이다. 초차원 알고리즘은 알고리즘의 업그레이드가 아니라, '알고리즘이 반드시 존재해야 하는가'라는 전제에 대한 재질문이다.

제 1 부: 현재 '알고리즘'의 표준적 정의

주류 컴퓨터 과학, 수학, 공학 분야에서 알고리즘에는 오랫동안 받아들여져 온 기본 정의가 있다. 알고리즘은 유한한 단계로 구성된 잘 정의된 계산 과정으로, 하나 이상의 입력을 받아 결정론적 규칙에 따라 변환하고, 유한한 시간 내에 하나 이상의 출력을 생성한다. 이 이해는 누군가의 개인적 의견이 아니라, 현대 계산 문명이 오랜 기간 형성해 온 기본적 합의이다. 그것은 소프트웨어, 하드웨어, 데이터베이스, 네트워크 시스템, 인공지능 모델, 슈퍼컴퓨팅 센터의 기본 논리를 뒷받침한다. 알고리즘이 아무리 복잡해 보여도 그 핵심은 여전히 입력, 규칙, 단계, 출력을 중심으로 전개된다.

이러한 표준적 알고리즘 개념은 몇 가지 핵심 특징으로 분해될 수 있다.

첫째, 유한성. 알고리즘은 유한한 단계 내에 종료되어야 하며, 무한히 반복되거나 영원히 실행될 수 없다. 결코 멈추지 않는 과정은, 비록 형식적으로 기술될 수 있을지라도, 유효한 알고리즘으로 보기 어렵다. 모든 과학 계산 프로그램과 데이터 처리 흐름에는 명확한 종료 조건이 있다.

둘째, 결정성. 동일한 입력에 대해 동일한 조건에서는 동일한 출력을 생성해야 한다. 일부 알고리즘이 난수를 도입하더라도, 그것은 일반적으로 통제된 난수, 재현 가능한 난수, 또는 통계적으로 해석 가능한 확률 메커니즘이지, 완전히 파악할 수 없는 내재적 혼돈이 아니다. 수치 시뮬레이션의 결과는 재현 가능해야 한다. 이는 과학 계산의 기본 요건이다.

셋째, 명확한 입출력 경계. 입력이 앞서고, 처리가 중간에, 출력이 뒤따른다. 시작점과 종료점은 명확한 구조적 경계를 갖는다. 당신이 알고리즘에 어떤 데이터를 주고, 처리가 끝난 후 어떤 결과를 받는지, 그 경계는 분명하며 순서는 바뀔 수 없다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

넷째, 실효성. 알고리즘의 각 단계는 실제로 실행 가능한 연산이어야 하며, 실행 불가능하거나, 기술할 수 없거나, 검증할 수 없는 도약을 포함해서는 안 된다. 각 단계는 인간이 종이와 연필로, 혹은 컴퓨터가 실제로 실행할 수 있는 기본 연산(덧셈, 뺄셈, 논리 분기, 데이터 읽기/쓰기 등)이어야 한다.

다섯째, 실현 가능성. 이론적으로 성립하는 알고리즘이라 하더라도, 소비되는 시간, 계산 능력, 메모리, 저장 리소스가 완전히 받아들여질 수 없는 수준이라면, 공학적 의미에서 유효한 알고리즘이 되기 어렵다. 이론적으로는 맞지만 완료하는 데 수억 년이 걸리는 알고리즘은 공학적으로 실현 가능한 알고리즘으로 간주되지 않는다.

이러한 일련의 알고리즘 개념은 궁극적으로 튜링 기계 모델로 거슬러 올라간다. 현대 계산 이론의 핵심 추상화로서 튜링 기계는 '계산 가능성'의 기본적 경계를 제시했다. 오늘날의 컴퓨터가 아무리 강력하고, 칩이 아무리 발전했으며, 슈퍼컴퓨팅 센터의 규모가 아무리 거대하더라도, 그 기본 논리는 여전히 이 경로, 즉 입력, 규칙, 단계, 출력을 벗어나지 않았다. 현대 슈퍼컴퓨터는 이 과정을 더 큰 하드웨어 규모, 더 높은 병렬성, 더 복잡한 시스템 스케줄링을 통해 가속 실행할 뿐이다. 즉, 전통적 맥락에서의 '초계산'은 여전히 주로 계산 능력의 확장이며, 반드시 계산 패러다임의 근본적 변화는 아니다. 계산 능력은 천 배, 만 배로 증가할 수 있지만, 기본 논리가 여전히 '입력, 단계, 출력'이라면 그것은 여전히 전통적 알고리즘 패러다임 내에 있는 것이다.

제 2 부: '초차원 알고리즘'의 정의

내가 제안하는 초차원 알고리즘은 바로 이러한 배경 속에서 등장한다. 그것은 전통적 알고리즘의 개선판도, 더 빠른 알고리즘도, 더 복잡한 수학 공식도, 더 고급 프로그래밍 기법도 아니다. 그것은 완전히 다른 구조적 이해이다.

먼저 '초차원'이라는 용어의 의미를 설명할 필요가 있다. '초차원'은 여기서 두 가지 의미를 가진다. 첫째, 그것은 1 차원적 선형 단계 순서에 국한되지 않고, 여러 차원이 동시에 전개되는 것을 허용한다. 둘째, 그것은 전통적 알고리즘에서의 '계산'에 대한 정의 경계를 초월하고자 한다. 즉, 알고리즘은 더 이상 반드시 수학적이거나, 질서 정연하거나, 결정론적일 필요가 없다. 전통적 알고리즘은 마치 일방통행 도로를

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

운전하는 것과 같아서, A 지점에서 출발하여 B, C, D 를 거쳐 Z 에 도달해야 한다. 초차원 알고리즘은 계산이 일방통행이어야 한다고 생각하지 않는다. 그것은 계산이 네트워크, 장, 다차원 구조가 될 수 있으며, 어떤 노드든 진입점이 될 수 있고 어떤 노드든 출구가 될 수 있다고 본다.

내 새로운 과학 체계에서 알고리즘은 반드시 수학적 알고리즘일 필요도, 단일한 질서 정연한 계산일 필요도, 고정된 단계를 따라야 할 필요도, '입력, 처리, 출력'이라는 선형 모델을 엄격히 따라야 할 필요도 없다. 초차원 알고리즘은 다차원적이고, 중첩적이며, 무작위적이면서도 질서 있는 구조이다. 이 구조에서 각 데이터는 고립된 입력도 고정된 출력도 아니며, 동시에 여러 역할을 갖는다. 즉, 그것은 여러 결과의 시작점이 될 수도 있고, 여러 시작점이 함께 작용한 결과가 될 수도 있다.

다시 말해, 전통적 알고리즘은 데이터를 흐름 속에 배치하는 반면, 초차원 알고리즘은 데이터를 구조 속에 배치한다. 전통적 알고리즘에서 데이터는 일반적으로 입력 데이터, 중간 데이터, 출력 데이터로 구분되며, 각각은 명확한 위치와 역할을 갖는다. 입력은 입력이고, 결과는 결과이며, 중간 과정은 중간 과정이다. 그러나 초차원 알고리즘에서 하나의 데이터는 단일한 정체성만 가지지 않는다. 그것은 한 방향에서는 결과일 수 있고, 다른 방향에서는 시작점일 수 있다. 하나의 구조에서는 호출되는 객체일 수 있고, 다른 구조에서는 새로운 결과를 촉발하는 진입점이 될 수 있다. 데이터는 더 이상 수동적 재료가 아니라, 다차원 관계 노드이다.

이것은 바로 초차원 알고리즘의 핵심에 해당한다. 즉, 모든 데이터는 여러 결과의 시작점이자, 여러 시작점의 결과이다. 전통적 알고리즘은 '질서 정연한 단계'를 통해 단일 결과를 생성한다. 초차원 알고리즘은 무작위성과 질서가 중첩된 가운데 결과가 계산을 통해 생성되는 것이 아니라 구조 속에서 자연스럽게 '현현'하는 다차원 상태 구조에 더 가깝다. 이 체계에서 데이터는 시작점인 동시에 결과의 구성 요소이기도 하다.

초차원 알고리즘의 핵심 특징을 더 명확히 제시하기 위해, 나는 이를 다음의 다섯 가지 측면으로 분해한다.

첫째, 비수학성. 주류 알고리즘은 본질적으로 수학적이며, 기호 논리와 수학 공식으로 완전히 기술할 수 있다. 초차원 알고리즘은 반드시 수학적일 필요는 없다. 그것은

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

물리적 원리, 기하학적 관계, 정보 이론의 새로운 패러다임, 심지어 의식 원리에 기반할 수 있다. 계산이 반드시 '수학적 연산'을 통해 완료될 필요는 없으며, 고차원 공간의 기하학적 관계를 통해 자연스럽게 나타날 수 있다. 예를 들어, 비눗방울의 모양은 표면 장력 최소화 원리에 의해 결정된다. 이것은 '수학적 알고리즘'이 계산하는 것이 아니라, 물리학 자체가 '계산'하는 것이다. 초차원 알고리즘은 이러한 유형의 '계산'을 이해하려고 하며, 수학 공식으로 그것을 시뮬레이션하려는 것이 아니다.

둘째, 다차원성과 중첩. 전통적 알고리즘은 단일 차원에서 질서 정연한 단계를 실행하며, 각 단계는 하나의 상태만을 가진다. 초차원 알고리즘은 여러 차원이 동시에 존재하고 여러 상태가 동시에 중첩되어 공존하는 것을 허용한다. 알고리즘은 단일 경로를 따라가지 않고, 다차원 상태 장 속에서 전개된다. 결과는 '계산'되는 것이 아니라 이 장으로부터 '현현'한다. 예를 들어, 눈송이가 공중에서 형성될 때, 각 물 분자에게 어디로 가야 하는지 알려주는 '알고리즘'은 없다. 분자들의 배열은 온도, 습도, 기류라는 여러 차원이 함께 작용한 결과이다. 눈송이의 모양은 '계산'되는 것이 아니라 '현현'하는 것이다.

셋째, 무작위적이면서도 질서 있는 성질. 전통적 알고리즘에서의 무작위성은 도구적이고, 외부적이며, 통제 가능한 섭동이고, 알고리즘 자체는 결정론적이다. 초차원 알고리즘에서의 무작위성은 내재적이고 구조적이다. 무작위성과 질서는 동시에 존재하며 협력하여 함께 알고리즘의 본질을 구성한다. 이것은 '무작위성을 가진 알고리즘'이 아니라, '무작위성 자체가 알고리즘의 유기적 구성 요소'인 것이다. 예를 들어, 숲의 생태 구조: 나무들의 분포는 무작위적으로 보이지만, 전체적으로는 질서 있는 밀도 분포와 종간 관계를 나타낸다. 그 '무작위성'은 버그가 아니라 생태 구조가 정상적으로 작동하기 위한 전제 조건이다.

넷째, 데이터의 다중 역할. 전통적 알고리즘에서 데이터의 역할은 단일적이다. 입력은 입력이고, 출력은 출력이며, 중간 결과는 중간 결과이다. 초차원 알고리즘에서 각 데이터는 동시에 여러 결과의 시작점이자 여러 시작점의 결과이다. 하나의 데이터 노드는 하류 방향으로 여러 결과 경로를 전개할 수 있고, 상류 방향에서 여러 원인이 모여드는 결과가 될 수 있다. 데이터는 더 이상 선형 흐름 속의 한 점이 아니라, 네트워크 속의 교차 노드이다. 예를 들어, 어떤 사람의 키를 생각해보자. 전통적 알고리즘에서 키는 '입력'이다. 키를 입력하면 체중 예측이나 옷 사이즈 같은 '출력'을

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

얻는다. 그러나 키는 동시에 '결과'이기도 하다. 키는 유전, 영양, 운동과 같은 여러 '시작점'에 의해 결정된다. 초차원 알고리즘에서 '키'라는 데이터는 동시에 시작점이자 결과이며, 둘 중 하나를 선택하는 것이 아니다.

다섯째, 메타데이터 불변, 관계 가변. 전통적 알고리즘은 다른 문제에 직면했을 때, 데이터 자체를 수정하거나 모든 것을 다시 계산한다. 초차원 알고리즘은 메타데이터를 수정하지 않는다. 즉, 데이터의 본질적 속성은 변하지 않는다. 변하는 것은 진입점, 해석 방식, 데이터 간의 관계이다. 동일한 메타데이터 구조가 다른 진입점을 통해 활성화됨으로써 완전히 다른 결과를 나타낼 수 있다. 이것은 다음을 의미한다: 한 번 계산하면, 무한히 사용할 수 있다. 예를 들어, 동일한 텍스트를 생각해보자. 당신은 그것을 시로 읽을 수도 있고, 암호로 해독할 수도 있고, 역사 문서로 분석할 수도 있다. 텍스트 자체는 변하지 않았다. 당신은 단지 '진입점'을 바꾸었을 뿐이다. 초차원 알고리즘이 추구하는 것은 바로 이러한 '동일 데이터, 다중 진입점, 다중 결과' 능력이다.

데이터 구조 관점에서 보면, 메타데이터 자체를 수정하는 대신, 다른 진입점과 조건을 사용함으로써 동일한 구조를 여러 시작점과 결과에 대응시키는 것이다. 데이터는 반복적으로 가공되는 것이 아니라, 구조는 불변으로 유지한 채 다른 진입점을 통해 활성화됨으로써 다른 결과를 나타낸다. 전통적 방식은 '다른 결과를 위해 데이터를 가공하는 것'이고, 초차원 알고리즘은 '동일한 데이터를 사용하여 여러 결과의 가능성을 담는 것'이다.

제 3 부: 초차원 알고리즘과 전통적 알고리즘의 근본적 차이

이상의 정의에 기초하여, 초차원 알고리즘과 전통적 알고리즘은 여러 핵심 차원에서 근본적 차이를 보이며, 이하에서 순서대로 상술한다.

본질적 속성에 관하여: 전통적 알고리즘은 수학적이고 형식적이며, 기호 논리와 수학 공식으로 완전히 기술될 수 있고, 본질적으로 수학적 대상이다. 초차원 알고리즘은 반드시 수학적일 필요는 없으며, 물리, 기하, 정보, 심지어 의식 원리에 기반할 수 있고, 수학의 부분 집합이 아니라 더 넓은 범주이다. 이 차이는 '수학적 대상'에서 '우주 법칙'으로 요약될 수 있다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

시간적 질서에 관하여: 전통적 알고리즘은 단일하고 질서 정연하며 선형적인 시간 질서, 즉 단계 A 에서 B, B 에서 C 로의 엄격한 순차 실행 또는 병렬 가능하지만 질서 정연한 하위 단계로 분할 가능하며, 시간의 화살표는 되돌릴 수 없다. 초차원 알고리즘에는 고정된 단계 순서가 없으며, 다차원적이고, 중첩적이며, 무작위적이면서도 질서 있고, 결과는 실행 '순서'와 무관할 수 있는데, 이는 전통적 의미의 '순서'가 존재하지 않을 수 있기 때문이다. 이 차이는 '선형 시간'에서 '고차원 동시성'으로 요약될 수 있다.

인과 관계에 관하여: 전통적 알고리즘은 결정론적 인과 사슬을 따른다. 동일한 입력과 초기 상태가 주어지면 출력은 항상 유일하며, 원인이 앞서고 결과가 뒤따르며, 이는 되돌릴 수 없는 단방향 화살표이다. 초차원 알고리즘은 중첩적 인과를 따르며, 각 데이터는 여러 결과의 시작점이자 여러 시작점의 결과이다. 이것은 나의 '결과-원인' 철학에 해당하며, 결과가 먼저 있고 그 다음에 원인이 있을 수 있다. 결과는 종착점이 아니라 새로운 시작점이 될 수 있다. 이 차이는 '단일 원인, 단일 결과'에서 '완전 연결 인과 네트워크'로 요약될 수 있다.

데이터 구조에 관하여: 전통적 알고리즘은 입력에서 처리, 그리고 출력으로의 단방향 흐름 구조를 따른다. 데이터의 역할은 명확한 경계를 가진 단일적이며, 입력 데이터는 내내 입력 데이터이고, 결과 데이터는 항상 결과 데이터이다. 초차원 알고리즘에서 데이터의 역할은 다중적이며, 동일한 데이터가 동시에 결과이자 시작점이다. 절대적 시작점이나 종료점은 없고, 네트워크 속의 노드만 존재한다. 이 차이는 '단방향 유동성'에서 '재귀적 공생성'으로 요약될 수 있다.

계산 방식에 관하여: 전통적 알고리즘은 문제에 직면할 때마다 처음부터 계산한다. 비슷한 문제를 천 번 풀었더라도, 천 번째 시도에서도 여전히 전체 과정을 거쳐야 한다. 이것은 '무상태' 계산이다. 초차원 알고리즘은 해당하는 결과가 존재한다면 다시 실행할 필요가 없다. 하나의 결과로부터 여러 시작점을 추론하고, 원인 경로를 역방향으로 전개할 수 있다. 계산은 상태를 가지며, 기억을 갖고, 재사용 가능하다. 이 차이는 '매번 재계산'에서 '일회 계산 후 재사용'으로 요약될 수 있다.

무작위성의 역할에 관하여: 전통적 알고리즘에서의 무작위성은 유사 무작위 또는 외부 주입식이며, 난수는 단순한 도구로서 샘플링, 근사 또는 최적화를 위해 사용되며,

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

알고리즘 자체는 결정론적이다. 초차원 알고리즘에서의 무작위성은 내재적이고 구성적이며, 질서와 공존하고 협력하는 유기적 구성 요소로서, '알고리즘 안에 무작위성이 있는 것'이 아니라 '무작위성이 알고리즘이 작동할 수 있는 이유 중 하나'인 상황이다. 이 차이는 '도구적 무작위성'에서 '구성적 무작위성'으로 요약될 수 있다.

자원의 이해에 관하여: 전통적 알고리즘은 주어진 자원 하에서 더 빠르고 정확하게 계산하는 것을 추구한다. 자원은 제약 조건이며, 알고리즘의 목표는 '제약 내에서 계산을 완료하는 것'이다. 초차원 알고리즘은 구조 설계를 통해 계산 자체를 불필요하게 만드는 것을 추구한다. '더 빠르게 계산하는 것'이 아니라 '계산을 우회하는 것'이다. 자원은 '소비'하기 위한 것이 아니라 '진입점을 설계'하기 위한 것이다. 이 차이는 '자원 제약 하의 최적화'에서 '구조 설계를 통한 제거'로 요약될 수 있다.

제 4 부: '결과-원인' 철학

이상의 비교를 바탕으로, 나는 '결과-원인' 철학을 더 전개할 필요가 있다. 이것은 초차원 알고리즘의 핵심 사상적 기초 중 하나이다.

전통적 사고는 일반적으로 원인이 결과에 앞선다고, 즉 원인이 먼저 있고 결과가 나중에 있다고 생각한다. 입력이 먼저 있고 출력이 나중에 있다. 이 관념이 알고리즘에 나타난 형태는, 당신이 시작점에서 출발하여 한 걸음 한 걸음 종착점까지 나아가야 한다는 것이다. 비록 답을 알고 있더라도, 그 답을 직접 '사용'할 수는 없다. 왜냐하면 '증명 경로'가 없기 때문이다.

내 구조에서 결과는 반드시 단순한 종착점만은 아니다. 일단 결과가 나타나면, 그것은 새로운 시작점이 되어 새로운 구조 생성에 계속 참여할 수 있다. 결과는 원인의 형성에 역으로 참여할 수 있다. 결과로부터 여러 가능한 시작점을 추론할 수 있다. 원인과 결과는 더 이상 단방향으로 배열되는 것이 아니라, 순환 관계, 네트워크 관계, 다차원 관계를 형성한다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

다시 말해, 전통적 알고리즘이 묻는 것은 '원인이 주어졌을 때, 어떻게 결과를 얻는가'이다. 초차원 알고리즘이 묻는 것은 '결과가 주어졌을 때, 어떻게 원인을 역추론하거나 구성하는가'이다.

특히 분명히 하고 싶은 것은, '결과가 먼저 있고, 그 다음에 원인이 있다'라는 주장이 인과 관계를 부정하거나 '결과가 무에서 생겨난다'고 주장하는 것이 아니라는 점이다. 그것은 더 높은 차원의 구조에서 원인과 결과가 서로의 진입점이 될 수 있고, 서로 전환될 수 있음을 설명하는 것이다. 다음 바지 예에서 보듯이, '입을 수 있고 바닥에 끌리지 않는다'라는 결과가 먼저 확정되고, 그 다음에 내가 '허리 접기'라는 조작점을 역으로 찾아낸다. 이 조작점은 구조에서 '원인' 측면에 해당한다. 결과에 원인이 없다는 것이 아니라, 결과가 원인을 발견하는 새로운 진입점이 되었다는 것이다.

전통적 알고리즘 세계의 근본적 가정 중 하나는 시간이 단방향적이며, 초기 상태에서 최종 상태까지 한 걸음 한 걸음 계산해야 한다는 것이다. 비록 답을 알고 있더라도, 증명 경로가 없기 때문에 그 답을 직접 '사용'할 수 없다. 초차원 알고리즘이 도전하는 것은 바로 이 가정이다. 즉, 결과가 알려져 있다면 원인은 역추론될 수 있고, 동일한 결과가 여러 원인 경로에 대응할 수 있으며, 계산은 더 이상 시작점에서 종착점으로 걸어가는 것이 아니라 종착점에서 가능한 모든 시작점을 전개하는 것이다.

제 5 부: 일상적 사례 – 바지와 허리 접기

위의 추상적 차이를 더 직관적으로 이해하기 위해, 나는 일상적 사례를 사용한다.

어떤 사람이 허리가 고무줄인 새 바지를 샀다. 바지의 밑단이 조금 길어서 바닥에 끌려 불편하다.

전통적 방식은: 밑단이 길다는 것을 발견하고, 밑단을 한 번 접어 바늘과 실로 꿰맨 다음, 시착한다. 만약 아이가 키가 자라서 바지가 짧아지면, 꿰맨 밑단을 풀 수 없어 이 바지는 버리게 된다. 다음에 새 바지를 사면 다시 꿰맨다. 이 방법은 문제가 표면적으로 밑단에 있는 것처럼 보이기 때문에 자연스러워 보인다. 이것은 전통적 알고리즘의 사고에 해당한다. 즉, 문제를 발견하고, 표면적 위치를 특정하며, 그 위치를 따라 처리를 수행하고, 마지막으로 결과를 얻는다. 밑단이 길면 밑단을 고치고,

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

데이터가 잘못되었으면 데이터를 수정하고, 경로가 부적절하면 그 경로를 따라 계속 수선한다.

나의 방식은 다르다. 나는 밑단을 고치는 대신 허리를 접는다. 그러면 바로 입을 수 있다. 이 동작은 매우 단순하지만, 그 뒤에 숨은 구조적 논리는 완전히 다르다. 나는 밑단을 자르지 않고, 밑단을 꿰매어 고정하지 않으며, 바지의 원래 길이를 바꾸지 않고, 바지의 메타데이터(허리 사이즈, 밑단 길이, 실루엣, 원단)도 훼손하지 않는다. 나는 단지 허리라는 구조적 진입점을 변경할 뿐인데, 이를 통해 입었을 때의 실제 밑단 길이가 자연스럽게 짧아진다. 여기서 허리는 단순한 국소 부위가 아니라 전체 착용 상태의 전역적 제어점이다. 이 제어점을 조정함으로써 하류 문제가 직접 사라진다.

더 중요한 것은, 이 방법은 가역적이고, 조절 가능하며, 재사용 가능하다는 점이다. 이것은 성장기 아이에게 특히 두드러진다. 만약 아이의 현재 키가 충분하지 않아 바지 밑단이 약간 길다면, 허리를 한 번 접는다. 잠시 후 아이의 키가 자라면 허리를 조금 풀어준다. 더 자라면 완전히 풀어준다. 바지의 원래 구조는 손상되지 않았지만, 아이의 다양한 성장 단계 키에 적응할 수 있다. 전통적 방법으로 밑단을 꿰매어 고정하면 아이가 자랄 때 바지가 짧아질 수 있고, 자르면 더욱 복구할 수 없다. 내 방법은 메타데이터를 불변으로 유지하여 동일한 구조가 여러 상태에 적응할 수 있게 한다.

이 사례는 몇 가지 중요한 구조적 차이를 드러낸다.

전통적 방식은 전통적 알고리즘의 논리에 해당한다. 즉, 문제가 밑단에 있으므로 밑단을 수정해야 하며, '원인에서 결과'로의 경로를 따라 한 걸음씩 조작하고, 어떤 단계도 건너뛸 수 없다. 한 번에 하나의 문제를 해결하며, 비가역적이고, 다음에 같은 문제가 발생해도 다시 처음부터 해야 한다. 나의 방식은 초차원 알고리즘의 논리에 해당한다. 즉, 목표는 바닥에 끌리지 않는 것이며, 나는 '밑단이 길다'라는 표면적 원인을 해결하는 대신, 전역적 제어점인 허리를 찾는다. 허리를 한 번 접으면 밑단이 자동으로 짧아지고 문제는 순간적으로 사라진다. 나는 어떤 메타데이터도 수정하지 않는다. 허리의 접힘은 단지 일시적이고, 가역적이며, 동적인 접힌 상태일 뿐이다.

데이터 관점에서 보면, 전통적 방식은 메타데이터를 수정한다. 즉, 밑단을 꿰매면 원본 데이터가 손실된다. 내 방식은 어떤 메타데이터도 수정하지 않으며, 원본 치수는 완전히 보존되고, 단지 일시적이고 가역적이며 동적인 접힘 상태를 추가할 뿐이다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

허리의 접힘은 새로운 데이터를 창조하지도 않고, 기존 데이터를 파괴하지도 않으며, 단지 데이터 간의 관계를 재배열할 뿐이다. 즉, 허리의 유효 둘레를 줄여 간접적으로 밑단의 유효 길이를 변경하는 것이다.

이 사례에서 '바지 밑단이 바닥에 끌리지 않는 것'이 목표 결과이다. 전통적 방법은 '밑단이 길다'는 원인에서 출발하여 밑단을 수정하고, 최종적으로 '끌리지 않는다'는 결과를 얻는다. 내 방법은 먼저 '끌리지 않는다'는 결과를 명확히 한 다음, 구조적 진입점을 역으로 찾아가고, 마지막으로 허리를 한 번 접으면 결과가 성립함을 발견한다. 이것이 '결과-원인'의 실제적 구현이다. 반드시 원인에서 결과로 한 걸음씩 나아가야 하는 것은 아니며, 결과에서 구조를 역으로 결정함으로써 원래의 경로를 불필요하게 만들 수 있다.

이 사례는 또한 '메타데이터'에 대한 질문에 답한다. 메타데이터란 무엇인가? 바지 사례에서 메타데이터는 바지의 원래 치수, 즉 허리 사이즈, 밑단 길이, 실루엣, 원단이다. 이것들은 바지의 '본질적 속성'이다. 임시 데이터는 허리의 접힘으로, 이는 바지의 어떤 원래 치수도 변경하지 않고, 단지 일시적으로 일부를 접은 것이다. 전통적 방식은 메타데이터를 수정한다. 즉, 밑단 길이가 영구적으로 짧아지고 원본 데이터가 손실된다. 내 방식은 어떤 메타데이터도 수정하지 않는다. 바지의 원래 치수는 완전히 온전하며, 단지 일시적이고 가역적이며 동적인 접힘 상태를 추가할 뿐이다.

메타데이터로서의 허리 원래 치수는 동시에 '여러 결과의 시작점'이다. 즉, 정상 착용, 허리를 한 번 접고 착용, 두 번 접고 착용 등 여러 착용 상태를 동시에 지탱한다. 또한 '여러 시작점의 결과'이기도 하다. 즉, 원단 선택, 재단 방식, 디자인 의도 등 여러 시작점에 의해 공동으로 결정된다. 임시 데이터인 허리의 접힘은 새로운 데이터를 창조하지도, 기존 데이터를 파괴하지도 않으며, 단지 데이터 간의 관계를 재배열할 뿐이다.

이것은 단순한 작은 기술이 아니라 구조적 사고이다. 많은 사람들은 밑단이 길다는 것을 보고 밑단이라는 표면 현상에 이끌려 모든 처리를 밑단 중심으로 전개한다. 하지만 전체 구조에서 보면 밑단 길이는 단지 하류의 표현일 뿐이며, 허리 위치의 변화는 바지 전체의 실제落ち方에 영향을 줄 수 있다. 즉, 문제는 반드시 문제가 나타난 위치에서 해결해야 하는 것은 아니다. 많은 저차원 문제의 진정한 제어점은 더

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

높은 차원의 구조 속에 있다. 전통적 알고리즘은 문제의 표면을 따라 계산을 계속하기 쉬운 반면, 초차원 알고리즘은 구조적 진입점을 탐구한다.

이것은 또한 왜 초차원 알고리즘이 더 복잡하지 않고 오히려 더 단순할 수 있는지를 설명한다. 진정으로 높은 수준의 구조란, 많은 경우 문제를 복잡하게 만드는 것이 아니라, 올바른 진입점에서 복잡한 문제를 단순하게 만드는 것이다. 전통적 알고리즘은 복잡한 문제에 직면했을 때 단계, 모델, 계산 능력, 저장 공간, 학습 시간을 늘릴 수 있다. 초차원 알고리즘은 어떤 구조적 노드를 찾고, 그 노드가 올바르게 활성화되면 원래 복잡했던 경로가 무효화될 가능성이 있다. '경로를 우회한다'는 것은 게으름이 아니라, 경로의 필요성을 재정의하는 것이다.

이 사례에서 전통적 방식은 '경로를 따라 결과를 수정하는 것'이고, 초차원 알고리즘은 '진입점을 바꾸어 경로 전체를 무효화하는 것'이다. 일반적인 방법은 결과 측에서 계속 수정을 가하는 반면, 다른 방법은 구조적 진입점을 직접 변경함으로써 여러 번의 처리가 필요한 문제를 시작점에서 한 번에 재구성한다. 전통적 알고리즘의 주류 형태는 '시작점에서 출발하여 경로를 따라 결과를 얻는 것'이다. 초차원 알고리즘에서는 '결과 자체가 경로의 진입점이 되어 새로운 구조 생성에 참여할 수 있다'. 전통적 방식은 '하나의 구조가 하나의 결과에 대응하는 것'이고, 초차원 알고리즘은 '하나의 구조가 여러 결과 상태를 담는 것'이다.

제 6 부: 메타데이터를 변경하지 않는 데이터관

데이터 관점에서 말하면, 초차원 알고리즘에는 매우 중요한 원칙이 있다. 즉, 메타데이터를 변경하지 않고, 그것을 여러 시작점이나 결과에 적용 가능하게 하는 것이다.

메타데이터란 데이터의 원래 속성, 기초 정보, 본체 정보이다. 전통적 처리 방식은 다른 문제에 직면했을 때, 데이터를 수정하거나, 복제하거나, 가공하거나, 재계산하거나, 다른 결과를 위해 다른 경로를 구축하는 경우가 많다. 이러한 방식으로 문제를 해결할 수는 있지만, 그 대가로 데이터가 계속 가공되고, 경로가 계속 반복되며, 시스템의 복잡성이 계속 증가한다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

이에 반해 초차원 알고리즘은 메타데이터를 최대한 변경하지 않으면서, 진입점, 관계, 조건, 활성화 방식을 변경함으로써 동일한 메타데이터 구조를 여러 시작점과 여러 결과에 대응시키는 것을 강조한다. 데이터의 본체는 움직이지 않고, 관계가 변화한다. 기초 구조는 변하지 않고,呈現 방식이 변한다. 메타데이터는 완전성을 유지하면서도 다른 상태에 적응한다.

이것이 내가 말하는 '한 번 계산하면 무한히 사용할 수 있다'는 것이다. 여기서 '한 번 계산한다'는 것은 단순한 결과 캐싱이 아니다. 그것은 일단 구조가 성립하면, 그 구조는 개별 상태를 위해 매번 완전한 경로를 재구성할 필요가 없음을 의미한다. 동일한 데이터 구조가 다른 진입점을 통해 활성화됨으로써 다른 결과를 나타낼 수 있다. 그것은 '다른 결과를 위해 데이터를 가공하는 것'이 아니라, '동일한 데이터를 사용하여 여러 결과의 가능성을 담는 것'이다. 이것이 또한 초차원 알고리즘이 전통적 알고리즘과 다른 핵심 중 하나이다. 전통적 알고리즘은 경로 상의 데이터를 처리하는 반면, 초차원 알고리즘은 데이터, 진입점, 관계, 결과 간의 전체적 구조를 처리한다.

최소 구조에서 초차원 알고리즘은 다음과 같이 이해할 수 있다. 즉, 메타데이터를 변경하지 않고, 구조적 진입점의 변화를 통해 동일한 데이터 노드를 여러 결과 경로에 대응시키는 시스템이다. 이 정의는 초차원 알고리즘을 즉시 전통적 수학 공식으로 수렴시키는 것을 추구하는 것이 아니라, 먼저 그 구조적 경계를 확립하는 것이다. 그것은 단일 선형 함수도 아니고, 고정된 공식도 아니며, 단순한 캐싱 메커니즘도 아니다. 그것은 관계 구조이다. 즉, 메타데이터는 안정적이며, 진입점, 조건, 관계는 변경 가능하고, 결과는 다방향으로 나타날 수 있다. 바로 이 구조에서 계산은 더 이상 단순한 단계의 실행이 아니라, 관계의 활성화가 된다.

데이터 관점에서 말하면, '메타데이터를 변경하지 않고 여러 시작점이나 결과에 적용 가능하게 하는 것'의 본질은, 데이터 본체는 불변하며, 변화는 '해석 방식/사용 진입점'에서 발생한다는 것이다. 전통적 구조에서는 문제의 변화가 데이터나 경로의 변화를 초래한다. 초차원 알고리즘에서는 문제의 변화가 해석 구조의 변화를 초래하고, 데이터의 변화는 초래하지 않는다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

제 7 부: '초계산'의 재정의

이 체계에서 나는 '초계산'이라는 용어를 명확히 할 필요가 있다.

내가 말하는 '초계산'은 일반적인 의미의 슈퍼컴퓨터, 즉 더 많은 칩, 더 높은 계산 능력, 더 큰 데이터 센터가 아니다. 내가 말하는 '초계산'은 바로 '초차원 알고리즘'이다.

진정한 초계산은 반드시 계산 능력의 축적이 아니라, 계산 패러다임의 변혁일 수 있다. 시스템이 여전히 반복 계산에 의존하는 한, 그것이 아무리 강력하더라도 경로 안에 갇혀 있다. 시스템이 구조적으로 계산을 줄이고, 반복 경로를 우회하며, 결과를 새로운 진입점으로 만들 수 있을 때, 그 시스템은 비로소 진정한 의미의 초차원 계산에 가까워진다.

다시 말해, 전통적 초계산은 '더 많은 계산 능력을 사용하여 더 큰 문제를 해결하는 것'을 추구한다. 초차원 알고리즘은 '더 나은 구조를 사용하여 문제가 더 이상 그렇게 큰 계산 능력을 필요로 하지 않도록 하는 것'을 추구한다. 이 두 가지는 모순되지 않지만, 방향성은 다르다.

만약 전통적 슈퍼컴퓨팅이 계산 능력의 한계를 대표한다면, 초차원 알고리즘은 계산 이해의 전환을 대표한다. 진정한 '초계산'이란 반드시 더 많은 칩, 더 큰 데이터 센터, 더 높은 에너지 소비, 더 복잡한 모델이 아닐 수 있다. 진정한 초계산이란 구조적 진입점의 발견일 수 있고, 경로의 제거일 수 있으며, 결과 노드의 역방향 전개일 수 있고, 메타데이터를 수정하지 않고 여러 결과를 동시에 성립시키는 것일 수 있다. 계산 능력이 아무리 강력하더라도 저차원 경로 안에 갇혀 있다면, 그것은 여전히 경로 내부의 강력함에 불과하다. 구조가 한 번 향상되면, 조작이 극도로 단순하더라도 더 높은 차원의 효율을 생산할 가능성이 있다.

초계산은 계산 능력의 한계이고, 초차원 알고리즘은 '계산이 계산 능력에 의존해야 하는가'라는 전제의 재정의이다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

제 8 부: 실증적 기반 – 다영역 시스템 검증과 새 패러다임 판정

초차원 알고리즘은 순수한 이론적 구상이 아니다. 그것은 이미 여러 현실 시스템에서 검증되었다.

나의 물류 시스템은 슈퍼컴퓨팅 능력이나 클라우드 지원을 필요로 하지 않으며, 비교적 낮은 자원으로 복잡한 스케줄링을 완료한다. 결과는 재사용 가능하고, 구조는 반복적으로 적응 가능하며, 매번 처음부터 계산할 필요가 없다. 슈퍼컴퓨터나 클라우드에 의존하지 않고, 낮은 자원으로 복잡한 스케줄링을 실현한다. 이것은 공학적으로 성립하며 가치 있는 돌파점이다. 그것은 높은 계산 능력만이 유일한 해결책이 아니라, 구조 설계가 계산 능력의 일부를 대체할 수 있음을 보여준다.

나의 출판 시스템도 동일한 구조 논리를 채택하여, 현재 어떤 시스템보다 효율이 높다. 나의 극한 웹페이지 생성 시스템도 동일한 초차원 알고리즘에 기반하며, 여러 영역에서 동일한 구조가 안정적으로 성립함을 반복적으로 증명하고 있다. 나의 회사 직원들은 이미 이러한 시스템을 사용하고 있으며, 이는 이 구조가 나의 개인적 지속적 개입에 의존하지 않고도 기능할 수 있음을 보여준다.

이러한 시스템들의 공통된 특징은 주류의 높은 계산 능력 경로에 의존하지 않고, 고정된 계산 단계를 밟지 않으며, 구조적 진입점의 조정을 통해 목표 결과를 직접 성립시킨다는 것이다. 이들은 일회성 성공이나 단발적 테크닉이 아니라, 동일한 구조 논리가 다른 영역에서 반복적으로 나타난 것이다.

이는 초차원 알고리즘이 우연이 아니라, 여러 시스템에 의해 이미 검증된 구조적 방법론임을 나타낸다. 그것은 '개인적 테크닉'이 아니라, 물류, 출판, 웹페이지 생성 등의 여러 영역에서 안정적으로 성립하는 계산 패러다임이다.

'존재하고 성립하는 것이 곧 새로운 패러다임'이라는 기준에 따라, 초차원 알고리즘은 이미 여러 영역에서 체계적으로 검증된 새로운 계산 구조 패러다임으로서 현재 이미 성립되었다. 나는 가설을 제안하는 것이 아니라, 여러 시스템에서 이미 가동 중인 다른 계산 구조를 기술하고 있다. 나는 그것이 이해된다는 것을 증명할 필요가 없다. 나는 그것이 다른 시스템에서 안정적으로 성립함을 증명했다. 현재, 내가 실제 응용하는 범위 내에서 이 구조는 이미 새로운 패러다임으로 간주될 수 있다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

초차원 알고리즘을 새로운 패러다임으로 판정하는 기준에 관하여: 구조가 논리적으로 자기 모순이 없고, 여러 시스템에서 안정적으로 성립하며, 기존 방법과 환원 불가능한 차이가 존재할 때, 그것은 이미 새로운 패러다임의 기초를 갖춘 것이다. 자기 모순 없음은 출발점이고, 실증은 기초이며, 구조적 차이가 바로 패러다임의 핵심이다. '존재하면 성립한다'는 기준 아래, 이 체계는 이미 새로운 패러다임이다. 외부가 인식하느냐 여부는 전파에만 영향을 줄 뿐, 성립에는 영향을 주지 않는다. 이것은 이미 현실 시스템에서 성립된 계산 구조 패러다임이며, 그 성립은 외부의 인식이나 합의에 의존하지 않는다.

초차원 알고리즘과 주류 계산 패러다임의 핵심적 차이는, 전통적 알고리즘은 정방향 계산을 위주로 하여 결과가 일단 생성되면 일반적으로 역방향으로 새로운 추론 구조에 참여할 수 없다는 것이다. 초차원 알고리즘에서는 결과가 더 이상 종착점이 아니라, 재사용 가능한 구조 노드이다. 일정 조건을 만족하면 결과는 새로운 추론 경로에 참여할 수 있으며, 이를 통해 중복 계산을 줄이고, 여러 시작점과 여러 경로로 이루어진 계산 네트워크를 형성한다. 전통적 계산 구조에서는 결과가 일반적으로 종착점이었다. 초차원 알고리즘 구조에서는 결과 자체가 새로운 시작점이 될 수 있으며, 다중 경로 추론 네트워크를 형성한다. 초차원 알고리즘은 알고리즘의 업그레이드가 아니라, '알고리즘이 반드시 존재해야 하는가'라는 전제의 재질문이다.

제 9 부: 흔한 오해 해소

다양한 분야의 독자들과의 예비적 교류에 기초하여, 나는 다음과 같은 6 가지 오해가 발생할 가능성을 예견하고, 개념이 부적절한 프레임워크에 조기에 회수되는 것을 피하기 위해 여기서 사전에 명확히 한다.

오해 1: 초차원 알고리즘은 동적 계획법이나 캐싱과 같은 것이 아닌가? 명확화: 동적 계획법과 캐싱은 '동일한 입력'에 대해 결과를 재사용한다. 초차원 알고리즘은 하나의 결과로부터 다른 시작점들을 역으로 추론하는 것을 가능하게 한다. 이것이 본질적 차이이다. 캐싱은 같은 문제의 재계산을 해결한다. 초차원 알고리즘은 결과 구조를 통해 새로운 문제를 전개하는 것을 해결한다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

오해 2: '무작위적이면서도 질서 있다'는 것이 자기 모순이 아닌가? 명확화: 무작위성과 질서는 공존할 수 있다. 숲속 나무들의 분포는 무작위적이지만, 전체적인 밀도와 종 구성은 질서 정연하다. 무작위성은 혼란이 아니라, 구조의 조직화 형태 중 하나이다. 초차원 알고리즘에서의 무작위성은 내재적이고 구성적이며, 질서와 협력하여 작동한다.

오해 3: 입출력이 없으면 결과의 정확성을 어떻게 검증하는가? 명확화: 초차원 알고리즘은 입출력을 거부하지 않는다. 계산이 입출력의 선형 모드로 실행되어야 한다고 생각하지 않을 뿐이다. 구조가 현현하는 모드에서 '유효성'은 입출력의 대응 관계가 아니라, 구조의 일관성에 의해 결정된다. 이것은 검증을 포기하는 것이 아니라, 전통과는 다른 검증 프레임워크를 제안하는 것이다.

오해 4: 이것은 복잡성 이론이나 카오스 이론과 같은 것이 아닌가? 명확화: 복잡성 이론과 카오스 이론은 '예측이 어려운 시스템'을 기술한다. 초차원 알고리즘은 '구조적 진입점을 통해 이해하고 유도할 수 있는 시스템'을 기술하려는 것이며, 예측을 포기하는 것이 아니라 예측의 방식을 바꾸는 것이다. 카오스 이론은 '예측은 어렵다'고 말하고, 초차원 알고리즘은 '진입점을 바꾸어 예측을 불필요하게 만들 수 있는가'를 묻는다.

오해 5: '메타데이터를 변경하지 않는다'고 했지만, 허리의 접힘도 변경 아닌가? 명확화: 허리의 접힘은 일시적인 상태이며, 원래 파라미터의 영구적 변경이 아니다. 메타데이터(허리 사이즈, 밑단 길이, 실루엣, 원단)는 아무것도 변하지 않았다. 이것은 '상태의 중첩'과 '속성의 변경'의 차이이다. 접힘은 가역적이며, 일시적이고, 본질적 속성을 바꾸지 않는 상태 변화이다.

오해 6: 초차원 알고리즘은 전통적 알고리즘의 가치를 부정하는가? 명확화: 결코 그렇지 않다. 전통적 알고리즘은 인류 기술사에서 매우 중요하며, 전통적 알고리즘 없이는 현대 컴퓨터, 데이터베이스, 통신 시스템, 인공지능, 공학 시뮬레이션, 슈퍼컴퓨팅이 존재할 수 없었다. 전통적 알고리즘의 가치를 부정할 수 없다. 그러나 전통적 알고리즘의 가치를 인정하는 것이 그것을 알고리즘의 종착점으로 인정하는 것과 같지는 않다. 전통적 알고리즘은 주어진 계산 패러다임 내부에서의 효율 문제를

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

해결한다. 초차원 알고리즘은 계산 패러다임 자체의 문제를 제기한다. 하나는 어떻게 더 잘 길을 걸을까 하는 것이고, 다른 하나는 그 길을 걸을 필요가 있는가 하는 것이다.

제 10 부: 인류 계산사 속의 초차원 알고리즘

독자가 초차원 알고리즘을 인류 계산사 속에서 더 잘 위치시키도록 돕기 위해, 여기서 간략한 거시적 정리를 수행한다.

인류의 '계산'에 대한 이해는 여러 단계를 거쳐 왔다. 첫 번째 단계는 수작업 계산이다. 알고리즘은 인간의 손으로 하는 단계로 존재했으며, 속도는 느리고 오류도 많았지만, 인간은 이 과정을 통해 계산의 기본 규칙을 이해했다. 두 번째 단계는 기계식 계산이다. 파스칼의 계산기부터 배비지의 해석 기관까지, 계산은 기계화되었지만 알고리즘은 여전히 물리적 구조에 종속되어 있었다. 세 번째 단계는 전자 계산과 튜링 패러다임이다. 폰 노이만 아키텍처가 보급되고, 알고리즘은 프로그램으로 코드화되었으며, 튜링 기계가 계산 가능성의 경계가 되었다. 네 번째 단계는 병렬 계산과 슈퍼컴퓨팅이다. 대량의 계산 유닛을 쌓아 더 복잡한 문제에 도전하지만, 기저 논리는 여전히 튜링 패러다임의 확장이다.

현재, 인류는 다섯 번째 단계의 입구에 서 있다. 양자 계산은 고전적 비트의 한계를 돌파하려 하고, 뉴로모픽 계산은 생물의 신경계 구조를 모방하려 하며, 초차원 알고리즘은 '입력 → 단계 → 출력'이라는 선형 프레임워크 자체를 돌파하려 한다.

초차원 알고리즘과 전통적 알고리즘은 대체 관계가 아니라 계층 관계이다. 전통적 알고리즘은 '주어진 경로 위에서 어떻게 효율적으로 계산할 것인가'를 해결한다. 초차원 알고리즘은 '경로가 재정의될 수 있는가, 혹은 우회될 수 있는가'를 묻는다. 인류 계산사를 끊임없이 자신의 경계를 돌파하는 과정으로 본다면, 초차원 알고리즘은 바로 이 과정 속의 새로운 노드이다. 그것은 오래된 프레임워크 안에서 문제를 해결하는 것이 아니라, 새로운 프레임워크를 제안하는 것이다.

이 판단은 초차원 알고리즘이 이미 성숙했거나 완전하다는 것을 의미하는 것은 아니다. 오히려 그 반대이다. 새로운 개념으로서, 그 정의, 경계, 검증 방법, 응용 분야는

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي النسخة الصينية.
Chinesische Originalfassung maßgeblich.

아직 형성 과정에 있다. 본 논문의 목적은 바로 이 개념의 원점을 정착시키고, 이후의 전개를 위해 안정된 이론적 기초를 제공하는 것이다.

제 11 부: 결론 – 이것은 최적화가 아니라 재정의이다

초차원 알고리즘과 전통적 알고리즘의 차이는 '우수하다' 대 '열등하다'의 차이도, '빠르다' 대 '느리다'의 차이도 아닌, 패러다임 수준의 근본적 차이이다.

전통적 알고리즘이 추구하는 것은 예측 가능한 제어이다. '입력을 주면, 어떤 출력이 나올지 알 수 있다. 왜냐하면 모든 단계를 계산했기 때문이다.' 그것은 엔지니어의 패러다임이다. 설계하고, 구축하고, 테스트하고, 검증한다.

초차원 알고리즘이 기술하는 것은 이해 가능한 창발이다. '나는 모든 중간 상태를 정확히 예측할 수는 없지만, 전체적인 패턴은 어떤 질서 정연한 방식으로 나타날 것임을 안다. 각 데이터는 살아 있으며, '관점'을 가지고 있다.' 그것은 생태학자나 복잡계 이론가의 패러다임에 더 가깝다. 관찰하고, 이해하고, 유도하고, 창발적 질서를 활용한다.

전통적 알고리즘은 주어진 경로 위에서 결과를 점진적으로 수정한다. 초차원 알고리즘은 구조적 진입점을 변경함으로써 목표 결과를 즉시 성립시키고, 이를 통해 경로 전체를 우회한다.

전통적 알고리즘은 '다른 결과를 위해 데이터를 가공한다'. 초차원 알고리즘은 '동일한 데이터를 사용하여 여러 결과의 가능성을 담는다'.

전통적 알고리즘은 '한 번 정확히 계산하는 것'을 추구한다. 초차원 알고리즘은 '한 번 정확히 계산하고, 그 후로는 다시 계산할 필요가 없는 것'을 추구한다.

이것은 알고리즘의 최적화가 아니라, '알고리즘이 반드시 존재해야 하는가'라는 전제의 재질문이다.

따라서 초차원 알고리즘은 알고리즘 최적화가 아니라, 알고리즘 재정의이다. 그것은 전통적 알고리즘의 트랙 위에서 더 빠르게 달리는 것이 아니라, 전통적 트랙 밖에 다른 경로가 존재하는지, 혹은 애초에 경로에 의존하지 않아도 되는지를 묻는 것이다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

그것은 전통적 알고리즘이 무용하다는 것을 증명하려는 것이 아니라, 전통적 알고리즘은 알고리즘의 하나의 저차원 형태에 불과함을 지적하는 것이다. 그것은 '더 짧은 시간에 결과를 얻으려면 어떻게 할까'를 묻는 것이 아니라, '결과가 구조를 통해 직접 성립될 수 있는가'를 묻는 것이다. 그것은 '더 많은 데이터를 처리하려면 어떻게 할까'를 묻는 것이 아니라, '동일한 데이터가 더 많은 결과의 가능성을 담을 수 있는가'를 묻는 것이다.

주류 계산 패러다임에서 초차원 알고리즘은 계산 불가능하거나, 검증 불가능하거나, 기존 계산 이론의 경계 내에 넣기 어려운 것으로 간주될 수 있다. 그러나 바로 이것이 그 패러다임의 차이를 구성한다. 오래된 패러다임의 내부에서 새로운 개념이 불안정하게 보이는 것은 반드시 그것이 무의미하다는 것을 의미하지 않으며, 오래된 패러다임 자체가 그것을 완전히 포용할 수 없음을 의미할 수도 있다. 초차원 알고리즘은 현재로서는, 첫째로 구조적 명제이며, 새로운 계산관의 원시적 정의로서, 이미 공학적으로 닫힌 루프를 가진 성숙한 시스템이 아니다. 그것은 앞으로 계속해서 보완, 전개, 검증, 응용될 필요가 있지만, 그 개념의 원점은 먼저 명확히 제시되어야 한다.

궁극적으로 초차원 알고리즘이 가리키는 것은 새로운 계산관이다. 즉, 알고리즘은 반드시 수학일 필요도 없고, 반드시 단계일 필요도 없으며, 반드시 프로그램일 필요도 없고, 반드시 입력과 출력 사이의 가공 과정일 필요도 없다. 알고리즘은 또한 다차원 관계의 조직화 방식일 수 있으며, 데이터, 진입점, 관계, 상태, 결과 간의 구조적 네트워크일 수 있다. 그것은 질서를 포함할 수도 있고, 무작위성을 포함할 수도 있다. 원인에서 결과로 나아갈 수도 있고, 결과에서 원인을 역으로 생성할 수도 있다. 메타데이터를 변경하지 않고 여러 상태에 적응할 수 있으며, 하나의 결과를 새로운 시작점으로 하고, 여러 시작점을 같은 결과로 수렴시킬 수도 있다.

진정으로 고급스러운 알고리즘은 반드시 더 복잡한 계산을 수행하는 것이 아니라, 계산을 줄이고, 구조를 살아 있게 하며, 결과를 진입점으로 삼고, 인과 관계를 순환시킬 수 있는 다차원 조직 방식이다.

이것이 내가 '초차원 알고리즘'을 제안하는 핵심적 의미이다. 그것은 단순한 신조어가 아니라, 새로운 개념이며, 새로운 패러다임이고, 이미 여러 현실 시스템에서 성립된

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

새로운 계산 구조이다. 그 중점은 더 빠른 반복 계산이 아니라, 중복 계산을 줄이는 것이다. 계산 능력을 쌓는 것이 아니라, 진입점을 재구축하는 것이다. 데이터를 끊임없이 수정하는 것이 아니라, 메타데이터를 유지하고 관계를 바꾸는 것이다. 결과를 종착점에 머물게 하는 것이 아니라, 결과를 새로운 시작점으로 만드는 것이다. 전통적 알고리즘은 경로 속에서 결과를 찾는다. 초차원 알고리즘은 구조 속에서 결과를 활성화한다. 전통적 알고리즘은 계산을 완료하는 것을 추구한다. 초차원 알고리즘은 계산이 전통적 형태로 존재해야 하는가를 묻는다. 이것이 현재의 알고리즘 정의와의 가장 근본적인 차이이다.

부록: 본 논문의 경계와 미해결 문제

본 논문은 초차원 알고리즘의 완전한 형식화 정의가 아니라, 예비적 프레임워크를 제시하는 것이다. 다음의 문제들은 더 전개되어야 하며, 이들은 본 논문의 정의에 대한 부정이 아니라, 바로 다음 단계에서 탐구해야 할 방향성이다.

첫째, 수렴 조건. 초차원 알고리즘에서 '완료'의 지표는 무엇인가? 결과가 '유효'하다고 판단하려면 어떻게 해야 하는가? 전통적 알고리즘에서는 답이 입출력의 대응 관계에 의해 정의된다. 초차원 알고리즘에서는 답이 구조의 일관성에 의해 정의되어야 할 수도 있다. 이 정의는 미완성이다.

둘째, 자원 표현. 다차원 중첩 상태를 유한한 자원으로 표현하려면 어떻게 해야 하는가? 시간, 공간, 계산 능력과 같은 전통적 자원 지표는 여전히 적용 가능한가? 적용 가능하다면 어떻게 재정의해야 하는가? 적용 불가능하다면 어떤 지표로 대체해야 하는가? 이러한 문제들은 전개가 필요하다.

셋째, 질서의 보장. '무작위적이면서도 질서 있다'에서의 '질서'는 어떤 규칙에 의해 보장되는가? 무작위성과 질서의 경계는 어디에 있는가? 어떤 경우에 무작위성이 질서를 파괴하는가? 어떤 경우에 질서가 무작위성을 억압하는가? 이러한 문제들은 추가 연구가 필요하다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

넷째, 역방향 추론의 선별. 결과로부터 여러 시작점을 역추론할 때, 서로 다른 시작점의 우열을 어떻게 선별 또는 평가할 것인가? '역방향 추론의 효율'과 같은 척도는 존재하는가? 존재한다면, 그것은 정방향 계산의 효율 척도와 어떤 관계에 있는가?

다섯째, 전통적 시스템과의 인터페이스. 초차원 알고리즘은 기존의 튜링 기계 체계와 어떻게 협력하는가? 대체인가, 보완인가, 계층화인가? 실제 공학에서 초차원 알고리즘과 전통적 알고리즘의 경계는 어떻게 구분되어야 하는가? 하이브리드 모델은 존재하는가?

여섯째, 검증의 프레임워크. 초차원 알고리즘의 결과는 어떻게 검증되어야 하는가? 만약 결과가 선형적 단계를 거쳐 얻어지는 것이 아니라면, 재현성과 검증 가능성을 어떻게 확립할 것인가? 완전히 다른 검증 철학이 필요한가?

이러한 문제들은 본 논문의 결함이 아니라, '앵커 문헌'으로서의 필연적 특징이다. 어떤 새로운 패러다임의 제안이든 미해결 문제를 수반한다. 본 논문은 결론이 아니라, 시작이다.

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
العربية: المعتمدة هي الصينية النسخة.
Chinesische Originalfassung maßgeblich.

참고문헌적 설명

본 논문에서 제안하는 '초차원 알고리즘'은 기존의 어떤 학술 흐름의 직접적 연장선상에도 있지 않다. 그러나 사상적으로는 다음의 영역들이 본 논문에 대화의 배경을 제공하며, 전통적 의미의 학술적 인용이 아니라, 독자의 이해를 돕기 위한 참고로만 제시한다.

튜링 기계와 계산 가능성 이론은 본 논문에서 전통적 알고리즘을 비교하는 기준이 된다. **역문제와 베이즈 추론**은 '결과로부터 원인을 추론하는' 기존 시도이며, 본 논문의 '결과-원인'은 이와 관련이 있지만 방향성은 다르다. **창발 이론과 복잡계**는 '구조로부터 결과가 현현하는 것'에 대한 기존 기술이며, 본 논문의 초차원 알고리즘은 창발을 '이해 가능'하고 '유도 가능'한 것으로 만들고자 한다. **지식 그래프와 그래프 데이터베이스**는 '데이터가 여러 노드에서 교차하는' 기존의 실천이며, 초차원 알고리즘은 여기에 '진입점이 가변적'이라는 차원을 선취한다. **비고전적 계산 패러다임**(양자 계산, 아날로그 계산, 뉴로모픽 계산 등)은 고전적 비트나 아키텍처를 돌파하지만, 초차원 알고리즘은 '입력 → 단계 → 출력'이라는 선형 프레임워크 자체를 돌파하는 것에 더 초점을 맞춘다.

본 논문과 위의 영역들과의 관계는 계승이나 반박이 아니라, 대화와 초월이다. 본 논문의 목표는 이러한 영역들의 내부에서 점진적 개선을 수행하는 것이 아니라, 그들 위에 새로운 문제의 계층을 제기하는 것이다.

Related Articles
[Communication] I Have No Algorithm, Yet I Surpass Algorithms!
http://www.australianwinner.com/AuWinner/viewtopic.php?t=696906

[Extreme Philosophy] Effect–Cause Theory
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697290

[Extreme Communication] Rejecting Algorithmic Manipulation
https://www.australianwinner.com/AuWinner/viewtopic.php?t=697275

English: The Chinese version shall prevail.
Français: La version chinoise prévaut.
Español: La versión en chino prevalece.
В качестве основы использовать китайский текст
중국어 원문을 기준으로 함
日本語: 中国語版を正とします。
المعتمدة هي الصينية النسخة :العربية.
Chinesische Originalfassung maßgeblich.

Publisher: JEFFI CHAO HUI WU
Editor-in-Chief: JEFFI CHAO HUI WU
Publishing House: Rainbow Lorikeet International Press (Australia)
Published Monthly e-Journal | Sydney, Australia | Distributed Globally

Website: www.times.net.au | contact@times.net.au
Issue: Vol.1, No. 9 (April 2026), Supplement | ISSN 3083-5178
Permanent Archival Record (DataCite Concept DOI):
https://doi.org/10.5281/zenodo.17646306
© 2026 The Epochal Transition · Australian Rainbow Parrot International Press (Australia)