THE AICE LIBRARY

The Examined Machine

เครื่องจักรที่ถูกไต่สวน

PHILOSOPHY AND COMPUTING, EACH READ THROUGH THE OTHER

★

Fifteen hearings on one shared question — what can be known, said, and done by rule — argued twice, by philosophy and by computing, until each discipline answers in the other's vocabulary.

สิบห้าการไต่สวนบนคำถามเดียวกัน — สิ่งที่รู้ได้ พูดได้ และทำได้ตามกฎ — ถูกโต้แย้งสองรอบ โดยปรัชญาและการคำนวณ จนแต่ละฝ่ายต้องตอบด้วยคำศัพท์ของอีกฝ่าย

2026-08-02 · FIFTEEN HEARINGS, TWO COURTS

How this book is built

Fifteen hearings, framed by a chapter that sets the courtroom and one that closes it: principles argued with dates, names, and sources, not a survey. Part I (chapters 01–08) puts computation on the stand and lets philosophy question it; Part II (chapters 09–15) reverses the benches. Read straight through to watch the case build, or open any hearing on its own — each chapter's first line places it in the two courts, and its Connections section says which chapters cross-examine it.

สิบห้าการไต่สวน กำกับด้วยบทที่วางห้องพิจารณาคดีไว้ตอนต้นหนึ่งบทและบทที่ปิดมันตอนท้าย อีกหนึ่งบท: อ้างหลักการด้วยวันที่ ชื่อบุคคล และแหล่งอ้างอิง ไม่ใช่การสำรวจ ภาค I (บทที่ 01–08) นำการคำนวณขึ้นที่นั่งพยานและให้ปรัชญาซักถาม ภาค II (บทที่ 09–15) สลับม้านั่งคู่ความ อ่านเรียงลำดับ เพื่อดูคดีค่อยๆ ก่อร่าง หรือเปิดอ่านการไต่สวนไหนก็ได้ทีละบท — บรรทัดแรกของทุกบทวางตำแหน่งของมันใน สองศาล และหัวข้อเชื่อมโยงบอกว่าบทไหนซักถามค้านบทนี้บ้าง

```
SSOT · books/philosophy/the-examined-machine/source/en/ (17 files)
TH contract · source/TH_SPEC.md
rebuild · core/_kit/venv/bin/python source/build_book.py
```

Contents

สารบัญ

CHAPTER 00

The Map

แผนที่

This chapter is the courtroom itself — the two benches, the rules of examination, and the standing of the witnesses. Every chapter after it is a hearing.

บทนี้คือห้องพิจารณาคดีเอง — ม้านั่งคู่ความสองฝั่ง กฎของการซักถาม และสถานะของพยาน ทุกบทหลังจากนี้คือการไต่สวนหนึ่งครั้ง

Why are philosophy and computing the same project, separated at birth?

ทำไมปรัชญากับการคำนวณจึงเป็นโครงการเดียวกัน ที่แยกทางกันตั้งแต่เกิด

For three days in September 1930, the limits of reason were argued twice in the same town by people who did not quite hear each other. The Second Conference on the Epistemology of the Exact Sciences met in Königsberg from 5 to 7 September; John von Neumann came to present the formalist case for securing mathematics by finite mechanical means (conference proceedings, *Erkenntnis* 2, 1931). On the last day, in a general discussion, Kurt Gödel let slip — off-hand, almost in passing — that he could exhibit arithmetical propositions that were true and unprovable in the formal system of classical mathematics. Von Neumann understood at once what had just been said (SEP, *Gödel's Incompleteness Theorems*). The next day, 8 September, a few streets away, David Hilbert closed an address to the Society of German Natural Scientists and Physicians with six words later cut into his tombstone: *Wir müssen wissen. Wir werden wissen.* We must know. We shall know.

> ตลอดสามวันในเดือนกันยายน 1930 ขีดจำกัดของเหตุผลถูกโต้แย้งสองครั้งในเมืองเดียวกัน โดยคนที่ไม่ค่อยได้ยินกันและกัน การประชุมครั้งที่สองว่าด้วยญาณวิทยาของวิทยาศาสตร์แม่นตรง (Second Conference on the Epistemology of the Exact Sciences) จัดขึ้นที่เคอนิกส์แบร์กระหว่างวันที่ 5 ถึง 7 กันยายน จอห์น ฟอน

> นอยมันน์มาเสนอข้อโต้แย้งแบบฟอร์มัลลิสม์ (formalism) เพื่อยืนยันความมั่นคงของคณิตศาสตร์ด้วยวิธีเชิงกลไกจำกัด (รายงานการประชุม, Erkenntnis 2, 1931) ในวันสุดท้าย ระหว่างการอภิปรายทั่วไป เคิร์ท เกอเดิลหลุดปากพูดขึ้นมาเฉยๆ เกือบจะเป็นเรื่องผ่านๆ ว่าเขาสามารถแสดงประพจน์ทางเลขคณิตที่เป็นจริงแต่พิสูจน์ไม่ได้ในระบบรูปนัย (formal system) ของคณิตศาสตร์คลาสสิก ฟอน นอยมันน์เข้าใจทันทีว่าเพิ่งเกิดอะไรขึ้น (สารานุกรมปรัชญาสแตนฟอร์ด (SEP), Gödel's Incompleteness Theorems) วันถัดมา คือ 8 กันยายน ห่างไปไม่กี่ถนน เดวิด ฮิลแบร์ทปิดปาฐกถาต่อสมาคมนักวิทยาศาสตร์ธรรมชาติและแพทย์แห่งเยอรมนีด้วยหกคำที่ภายหลังถูกสลักไว้บนหลุมศพของเขา: Wir müssen wissen. Wir werden wissen. เราต้องรู้ เราจะรู้

The remark and the vow were about the same thing, and the remark won. Hilbert's program was not arithmetic housekeeping; it was a philosophical wager — that every mathematical question is decidable, that certainty can be made mechanical (SEP, *Hilbert's Program*). Gödel published in 1931 and the wager was lost. Five years later the rest of the answer arrived, and it arrived in the shape of a machine.

> คำพูดหลุดปากกับคำปฏิญาณนั้นพูดถึงเรื่องเดียวกัน และคำพูดหลุดปากเป็นฝ่ายชนะ โครงการของฮิลแบร์ทไม่ใช่งานบ้านทางเลขคณิต แต่เป็นการพนันเชิงปรัชญา — ว่าทุกคำถามทางคณิตศาสตร์ตัดสินได้ (decidable) ว่าความแน่นอนทำให้เป็นกลไกได้ (SEP, Hilbert's Program) เกอเดิลตีพิมพ์ผลงานในปี 1931 และการพนันนั้นก็แพ้ ห้าปีต่อมาส่วนที่เหลือของคำตอบก็มาถึง และมันมาถึงในรูปของเครื่องจักร

That project was already two and a half centuries old. Gottfried Wilhelm Leibniz spent his life on a device and a dream: a *characteristica universalis*, a notation in which every concept decomposes into primitives, joined to a *calculus ratiocinator*, a rule-system for computing with them. The point was not tidiness but peace. "The only way to rectify our reasonings," he wrote, "is to make them as tangible as those of the Mathematicians, so that we can find our error at a glance, and when there are disputes among persons, we can simply say: Let us calculate, without further ado, to see who is right"

(Wiener, *Leibniz: Selections*, 1951, p. 51). The Latin is one word: *calculemus*. He also built the hardware: a hand-cranked calculator that multiplied and divided, one specimen of which surfaced in a Göttingen attic in 1879.

> โครงการนั้นมีอายุสองศตวรรษครึ่งอยู่ก่อนแล้ว กอทท์ฟรีท วิลเฮ็ล์ม ไลบ์นิซใช้ชีวิตทั้งชีวิตไปกับอุปกรณ์ชิ้นหนึ่งและความฝันหนึ่ง: characteristica universalis ระบบสัญลักษณ์ที่ทุกมโนทัศน์แยกย่อยลงเป็นหน่วยพื้นฐาน ผนวกเข้ากับ calculus ratiocinator ระบบกฎสำหรับคำนวณด้วยหน่วยเหล่านั้น ประเด็นไม่ใช่ความเป็นระเบียบ แต่คือสันติภาพ เขาเขียนไว้ว่า หนทางเดียวที่จะแก้ไขการให้เหตุผลของเราคือทำให้มันจับต้องได้เหมือนของนักคณิตศาสตร์ เพื่อที่เราจะเห็นความผิดพลาดของเราได้ในพริบตา และเมื่อเกิดข้อพิพาทระหว่างบุคคล เราก็เพียงพูดได้ว่า มาคำนวณกันเถิด โดยไม่ต้องพูดพร่ำต่อไป เพื่อดูว่าใครถูก (Wiener, Leibniz: Selections, 1951, p. 51) ภาษาละตินคือคำเดียว: calculemus เขายังสร้างฮาร์ดแวร์ด้วย: เครื่องคิดเลขแบบหมุนมือที่คูณและหารได้ ตัวอย่างหนึ่งของมันโผล่ขึ้นมาในห้องใต้หลังคาที่เกิททิงเงินในปี 1879

The line from that dream to the machine on your desk is not a clean one. George Boole did not learn that Leibniz had anticipated parts of his algebra until more than a year after *The Laws of Thought* (1854), when Robert Leslie Ellis pointed it out. Frege, by contrast, invoked the *calculus ratiocinator* by name and claimed the inheritance deliberately (SEP, *Leibniz's Influence on 19th Century Logic*). The dream was rediscovered at least as often as it was handed down — itself evidence for this book's claim: two disciplines keep arriving at the same idea because they are working the same seam.

> เส้นทางจากความฝันนั้นมาถึงเครื่องจักรบนโต๊ะทำงานของคุณไม่ใช่เส้นตรง จอร์จ บูลไม่รู้ว่าไลบ์นิซได้คาดการณ์บางส่วนของพีชคณิตของเขาไว้ล่วงหน้า จนกระทั่งกว่าหนึ่งปีหลังจาก The Laws of Thought (1854) เมื่อโรเบิร์ต เลสลี เอลลิสชี้ให้เห็น ในทางกลับกัน เฟรเกอ้างถึง calculus ratiocinator โดยตรงและอ้างสิทธิ์มรดกนั้นอย่างจงใจ (SEP, Leibniz's Influence on 19th Century Logic) ความฝันนี้ถูกค้นพบใหม่บ่อยพอๆ กับที่ถูกส่งต่อกันมา — ซึ่งเป็นหลักฐานของข้อเสนอหลักของหนังสือเล่มนี้เอง: สองสาขาวิชายังคงมาถึงความคิดเดียวกันซ้ำแล้วซ้ำเล่า เพราะทั้งคู่กำลังขุดตะเข็บเดียวกัน

The seam runs like this. Frege's *Begriffsschrift* (Halle, 1879) supplied quantifiers, and with them a logic strong enough to carry mathematical reasoning rather than merely classify syllogisms. Russell's letter of 16 June 1902 showed that Frege's foundation contained a contradiction. Hilbert and Ackermann's *Grundzüge der theoretischen Logik* (1928) posed the *Entscheidungsproblem*: is there a procedure that decides, for any statement of first-order logic, whether it is valid? Gödel answered a neighbouring question in 1931 with a no. Then in 1936 the *Entscheidungsproblem* itself fell twice, independently: Alonzo Church in April (*American Journal of Mathematics* 58: 345–363), and Alan Turing in a paper received 28 May.

> ตะเข็บนั้นเดินแบบนี้ Begriffsschrift ของเฟรเก (Halle, 1879) ให้ตัวบ่งปริมาณ (quantifier) มา และด้วยสิ่งนั้น ตรรกศาสตร์ก็แข็งแรงพอจะแบกรับการให้เหตุผลทางคณิตศาสตร์ ไม่ใช่แค่จำแนกซิลโลจิซึม (syllogism) จดหมายของรัสเซลล์ลงวันที่ 16 มิถุนายน 1902 แสดงให้เห็นว่ารากฐานของเฟรเกมีข้อขัดแย้งซ่อนอยู่ Grundzüge der theoretischen Logik (1928) ของฮิลแบร์ทกับอัคเคอร์มันน์ตั้งคำถาม Entscheidungsproblem (ปัญหาการตัดสินใจ) ขึ้นมา: มีกระบวนวิธีหนึ่งที่ตัดสินได้หรือไม่ ว่าข้อความใดๆ ในตรรกศาสตร์อันดับหนึ่ง (first-order logic) นั้นสมเหตุสมผลหรือไม่ เกอเดิลตอบคำถามข้างเคียงในปี 1931 ด้วยคำว่าไม่ได้ จากนั้นในปี 1936 ตัว Entscheidungsproblem เองก็ล้มลงสองครั้งโดยอิสระจากกัน: อลอนโซ เชิร์ชในเดือนเมษายน (American Journal of Mathematics 58: 345–363) และอลัน ทัวริงในบทความที่ได้รับเมื่อ 28 พฤษภาคม

Turing's method is the hinge of this whole book. To prove that no procedure exists, he first had to say what a procedure *is*, and "mechanical" had never been formally defined. So he defined it: an idealized clerk with a paper tape, a finite set of states, and a table of rules, and argued that this exhausts what any human computer following fixed instructions could do. The universal machine, and therefore the stored-program computer, is a by-product of

that argument. Nobody was trying to invent an industry. Someone was trying to settle whether reason has a decision procedure, and the computer fell out of the proof.

> วิธีของทัวริงคือบานพับของหนังสือทั้งเล่มนี้ เพื่อพิสูจน์ว่าไม่มีกระบวนวิธีใดอยู่จริง เขาต้องบอกก่อนว่ากระบวนวิธีคืออะไร และคำว่า "เชิงกลไก" ไม่เคยถูกนิยามอย่างเป็นทางการมาก่อน เขาจึงนิยามมันขึ้นมาเอง: เสมียนในอุดมคติที่มีเทปกระดาษ ชุดสถานะจำนวนจำกัด และตารางกฎ แล้วโต้แย้งว่าสิ่งนี้ครอบคลุมทุกสิ่งที่มนุษย์ผู้คำนวณตามคำสั่งตายตัวจะทำได้ เครื่องสากล (universal machine) และด้วยเหตุนี้คอมพิวเตอร์แบบโปรแกรมเก็บในตัว (stored-program computer) จึงเป็นผลพลอยได้จากข้อโต้แย้งนั้น ไม่มีใครกำลังพยายามสร้างอุตสาหกรรมขึ้นมา มีคนหนึ่งกำลังพยายามยุติคำถามว่าเหตุผลมีกระบวนวิธีตัดสินใจหรือไม่ แล้วคอมพิวเตอร์ก็หล่นออกมาจากบทพิสูจน์นั้น

The estrangement came fast, and it was institutional rather than intellectual. Turing published the founding paper of computing in a mathematics journal, then in October 1950 put the founding question of machine minds in *Mind*, a philosophy journal — one man, two faculties, fourteen years apart. By the time the first Department of Computer Sciences in the United States was created at Purdue in October 1962, it was established inside the Division of Mathematical Sciences: the new field took the machine from mathematics and left the question with philosophy (Rice & Rosen, Purdue CS history). Departments are cheap to found and expensive to bridge.

> ความแปลกแยก (estrangement) มาถึงอย่างรวดเร็ว และมันเป็นเรื่องเชิงสถาบันมากกว่าเชิงปัญญา ทัวริงตีพิมพ์บทความก่อตั้งวิชาการคำนวณในวารสารคณิตศาสตร์ แล้วในเดือนตุลาคม 1950 ก็นำคำถามก่อตั้งเรื่องจิตของเครื่องจักรไปลงใน Mind วารสารปรัชญา — คนคนเดียว สองคณะ ห่างกันสิบสี่ปี เมื่อภาควิชาวิทยาการคอมพิวเตอร์ (computer science) แห่งแรกในสหรัฐฯ ก่อตั้งขึ้นที่พูร์ดูในเดือนตุลาคม 1962 มันถูกตั้งไว้ภายในฝ่ายวิทยาศาสตร์คณิตศาสตร์: สาขาใหม่นี้เอาเครื่องจักรไปจากคณิตศาสตร์ และทิ้งคำถามไว้กับปรัชญา (Rice & Rosen, Purdue CS history) ภาควิชาก่อตั้งได้ง่าย แต่เชื่อมโยงกันยาก

Each side then forgot something the other was holding. **Computing forgot meaning.** Claude Shannon opened information theory in 1948 by setting semantics aside — "These semantic aspects of communication are irrelevant to the engineering problem" — an honest bracket, exactly right for building telephone networks. But a bracket held for eighty years hardens into a metaphysics, and a field that measures information without meaning eventually builds systems whose behavior it can quantify and cannot interpret. **Philosophy forgot feasibility.** Its standard epistemic agent knows every consequence of what it knows — logical omniscience, a creature no resource bound permits. Whole arguments about knowledge, rationality, and belief were built on an agent that could not be implemented at any scale (Aaronson, *Why Philosophers Should Care About Computational Complexity*, 2011).

> แต่ละฝ่ายลืมสิ่งที่อีกฝ่ายถืออยู่ **การคำนวณลืมความหมาย** โคลด แชนนอนเปิดทฤษฎีสารสนเทศในปี 1948 ด้วยการกันความหมาย (semantics) ออกไปก่อน — "แง่มุมเชิงความหมายของการสื่อสารเหล่านี้ไม่เกี่ยวข้องกับปัญหาเชิงวิศวกรรม" — เป็นการกันไว้อย่างซื่อตรง ถูกต้องเป๊ะสำหรับการสร้างเครือข่ายโทรศัพท์ แต่การกันไว้ที่ยืนยาวถึงแปดสิบปีย่อมแข็งตัวกลายเป็นอภิปรัชญา และสาขาที่วัดสารสนเทศโดยไม่มีความหมายในที่สุดก็สร้างระบบที่ตัวมันวัดพฤติกรรมได้แต่ตีความไม่ได้ **ปรัชญาลืมความเป็นไปได้จริง** ผู้รู้ (epistemic agent) มาตรฐานของมันรู้ทุกผลสืบเนื่องของสิ่งที่มันรู้ — ความรอบรู้เชิงตรรกะ (logical omniscience) สิ่งมีชีวิตที่ไม่มีขอบเขตทรัพยากรใดอนุญาตให้เป็นได้ ข้อโต้แย้งทั้งหมดเรื่องความรู้ เหตุผล และความเชื่อถูกสร้างขึ้นบนผู้รู้ที่ไม่มีทางนำไปใช้จริงได้ในสเกลใดเลย (Aaronson, Why Philosophers Should Care About Computational Complexity, 2011)

So this book is not a survey with two halves. It is a cross-examination, and the difference matters. A survey reports what each side believes; a cross-examination makes each side answer in the other's vocabulary, where its habits give it no protection. Three rules hold here. The examiner sets the terms — a witness allowed to answer in its own words is being interviewed, not examined. Both voices appear in every chapter: where the machine is on the stand it answers back, and where philosophy is on the stand it ob-

jects. And every claim carries its evidence — a date, a work, a source you can open — because the fastest way to lose a cross-examination is to assert what the record will not support.

> ดังนั้นหนังสือเล่มนี้ไม่ใช่การสำรวจที่แบ่งเป็นสองซีก มันคือการซักถามค้าน (cross-examination) และความต่างนี้สำคัญ การสำรวจรายงานว่าแต่ละฝ่ายเชื่ออะไร ส่วนการซักถามค้านบังคับให้แต่ละฝ่ายตอบด้วยคำศัพท์ของอีกฝ่าย ที่ซึ่งความเคยชินของตัวเองไม่อาจปกป้องมันได้ กฎสามข้อยืนอยู่ตรงนี้ ผู้ซักถาม (examiner) เป็นผู้กำหนดเงื่อนไข — พยาน (witness) ที่ได้รับอนุญาตให้ตอบด้วยถ้อยคำของตัวเองคือกำลังถูกสัมภาษณ์ ไม่ใช่ถูกซักถาม ทั้งสองเสียงปรากฏในทุกบท: ตรงที่เครื่องจักรอยู่ในที่นั่งพยาน (on the stand) มันตอบโต้กลับ และตรงที่ปรัชญาอยู่ในที่นั่งพยาน มันคัดค้าน และทุกข้อกล่าวอ้างแบกหลักฐานของตัวเองไว้ — วันที่ ผลงานชิ้นหนึ่ง แหล่งอ้างอิงที่เปิดดูได้ — เพราะทางที่เร็วที่สุดที่จะแพ้การซักถามค้านคือการยืนยันสิ่งที่บันทึกคดี (the record) ไม่รองรับ

Part I, chapters 01–08, puts the machine on the stand and lets philosophy ask: what kind of thing is a program, does a machine's answer count as knowledge, how does code mean, can a decision be delegated without losing its morality. Part II, chapters 09–15, reverses the benches: computation asks philosophy what it can actually deliver, and the answers often embarrass positions that sounded secure. No verdict is handed down at the end. Chapter 15 explains why the examination cannot close, and chapter 99 hands you the route to keep running it yourself.

> ภาค I บทที่ 01–08 นำเครื่องจักรขึ้นที่นั่งพยานและปล่อยให้ปรัชญาถาม: โปรแกรมคือสิ่งชนิดไหนกันแน่ คำตอบของเครื่องจักรนับเป็นความรู้หรือไม่ โค้ดมีความหมายได้อย่างไร การตัดสินใจถูกมอบหมายไปได้โดยไม่สูญเสียศีลธรรมของมันหรือไม่ ภาค II บทที่ 09–15 ม้านั่งสลับฝั่ง: การคำนวณถามปรัชญาว่ามันส่งมอบอะไรได้จริงบ้าง และคำตอบมักทำให้จุดยืนที่ฟังดูมั่นคงต้องอับอาย ไม่มีการอ่านคำตัดสิน (verdict) เมื่อจบ บทที่ 15 อธิบายว่าทำไมการซักถามจึงปิดไม่ได้ และบทที่ 99 ส่งมอบเส้นทางให้คุณไปวิ่งต่อเอง

How to read: seventeen files, each self-contained, principles rather than coverage. Every chapter opens with one line placing it in the two courts and closes with **Connections** — which chapters cross-examine it — plus open questions and a short, checkable reading list. Read straight through, or enter anywhere; the Connections lines are the corridors between courtrooms.

> วิธีอ่าน: สิบเจ็ดไฟล์ แต่ละไฟล์สมบูรณ์ในตัวเอง เน้นหลักการมากกว่าความครอบคลุม ทุกบทเปิดด้วยหนึ่งบรรทัดที่วางตำแหน่งของมันในสองศาล (the two courts) และปิดท้ายด้วย "เชื่อมโยง" — บทไหนซักถามค้านบทนี้บ้าง — บวกกับคำถามที่ยังเปิดอยู่และรายการอ่านสั้นๆ ที่ตรวจสอบได้ อ่านรวดเดียวจากต้นจนจบหรือเข้ามาตรงไหนก็ได้ บรรทัด "เชื่อมโยง" คือทางเดินเชื่อมระหว่างห้องพิจารณาคดี (courtroom) ต่างๆ

Connections · เชื่อมโยง

Everything connects here and this chapter stands on nothing. Chapter 01 is the closest neighbour — logic is where the two disciplines were still one family, and where Gödel, Turing, and Curry–Howard get handled properly. Chapter 09 returns to 1936 to ask what the Church–Turing thesis actually is: mathematics, empirical claim, or definition. Chapter 10 turns philosophy's forgotten feasibility into a standing objection. Chapter 11 audits Shannon's bracket in full.

> ทุกอย่างเชื่อมโยงมาที่นี่ และบทนี้ไม่ได้ยืนอยู่บนสิ่งใดมาก่อน บทที่ 01 คือเพื่อนบ้านที่ใกล้ที่สุด — ตรรกศาสตร์คือจุดที่สองสาขายังเป็นครอบครัวเดียวกัน และเป็นจุดที่เกอเดิล ทัวริง และ Curry–Howard ถูกจัดการอย่างถูกต้อง บทที่ 09 ย้อนกลับไปปี 1936 เพื่อถามว่าข้อตั้ง Church–Turing คืออะไรกันแน่: คณิตศาสตร์ ข้อกล่าวอ้างเชิงประจักษ์ หรือคำนิยาม บทที่ 10 เปลี่ยนความเป็นไปได้จริงที่ปรัชญาลืมไปให้กลายเป็นข้อคัดค้านที่ยืนหยัดถาวร บทที่ 11 ตรวจสอบการกันไว้ของแชนนอนอย่างละเอียดครบถ้วน

Open questions · คำถามที่ยังเปิดอยู่

Was the estrangement necessary, or merely administrative? A discipline that had kept its philosophers might have reached interpretability research decades earlier — or might have argued instead of building. Is the courtroom the right shape at all, or does an adversarial framing flatter both sides by assuming they are separable? And if the two are one project, why has no institution managed to house them together for long?

> ความแปลกแยกนั้นจำเป็นจริงหรือเป็นแค่เรื่องเชิงบริหาร สาขาวิชาที่รักษานักปรัชญาของตัวเองไว้อาจไปถึงงานวิจัยด้าน interpretability ได้เร็วกว่านี้หลายสิบปี — หรืออาจเอาแต่ถกเถียงแทนที่จะลงมือสร้าง ห้องพิจารณาคดีเป็นรูปแบบที่ถูกต้องเลยหรือไม่ หรือกรอบแบบปรปักษ์นี้กลับยกยอทั้งสองฝ่ายด้วยการสมมติว่าแยกจากกันได้ และถ้าทั้งสองเป็นโครงการเดียวกันจริง ทำไมไม่มีสถาบันไหนดูแลให้ทั้งคู่อยู่ด้วยกันได้นานเลย

Go deeper · อ่านต่อ

- Martin Davis, *The Universal Computer: The Road from Leibniz to Turing*, W. W. Norton, 2000 (paperback retitled *Engines of Logic*, 2001) — this chapter's arc at book length, told through the people.

  > Martin Davis, The Universal Computer: The Road from Leibniz to Turing, W. W. Norton, 2000 (ฉบับปกอ่อนเปลี่ยนชื่อเป็น Engines of Logic, 2001) — เนื้อหาของบทนี้ในความยาวระดับหนังสือทั้งเล่ม เล่าผ่านตัวบุคคล

- Alan M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, ser. 2, 42 (1936–37): 230–265 — doi.org/10.1112/plms/s2-42.1.230. Read alongside "Computing Machinery and Intelligence," *Mind* LIX, no. 236 (October 1950): 433–460 — doi.org/10.1093/mind/LIX.236.433 — for the split running through a single mind.

> Alan M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," Proceedings of the London Mathematical Society, ser. 2, 42 (1936–37): 230–265 — doi.org/10.1112/plms/s2-42.1.230 อ่านคู่กับ "Computing Machinery and Intelligence," Mind LIX, no. 236 (October 1950): 433–460 — doi.org/10.1093/mind/LIX.236.433 — เพื่อดูความแยกที่วิ่งผ่านจิตใจคนคนเดียว

— *The Philosophy of Computer Science*, Stanford Encyclopedia of Philosophy — the standing map of the borderland; pair it with the entries on the Church–Turing thesis and Hilbert's program.

> The Philosophy of Computer Science, SEP — แผนที่ประจำของดินแดนชายแดนนี้ อ่านคู่กับรายการเรื่องข้อตั้ง Church–Turingและโครงการของฮิลแบร์ท

CHAPTER 01

Logic

ตรรกศาสตร์

This is Part I's first hearing, the discipline standing closest to the courtroom itself — logic is where philosophy and computing were still, for a while, one undivided family.

นี่คือการไต่สวนแรกของภาค I สาขาวิชาที่ยืนใกล้ห้องพิจารณาคดีเองที่สุด — ตรรกศาสตร์คือจุดที่ปรัชญากับการคำนวณยังเป็นครอบครัวเดียวกันไม่แตกแยก อยู่ชั่วขณะหนึ่ง

What happened when the laws of thought became a machine?

เกิดอะไรขึ้นเมื่อกฎแห่งความคิดกลายเป็นเครื่องจักร

On 10 August 1937, a twenty-one-year-old graduate student named Claude Shannon submitted a master's thesis to MIT's electrical engineering department. Its claim was almost embarrassingly simple to state: the algebra George Boole had invented in 1854 to describe human reasoning — an algebra of true and false, of "and," "or," and "not" — was also the algebra of switches wired in series and parallel. A closed switch is 1; an open switch is 0. Two switches in series compute AND; two in parallel compute OR. Shannon showed that any relay circuit could be simplified the way a logician simplifies a formula, and that any Boolean formula could, in turn, be built out of relays. The published version, "A Symbolic Analysis of Relay and Switching Circuits" (*Transactions of the American Institute of Electrical Engineers* 57, 1938: 713–723), led the psychologist Howard Gardner, in 1985, to call it "possibly the most important, and also the most famous, master's

thesis of the century" (History of Information). Logic had been a description of thought for two thousand years. After Shannon, it was also a wiring diagram.

> วันที่ 10 สิงหาคม 1937 นักศึกษาปริญญาโทวัยยี่สิบเอ็ดปีชื่อโคลด แชนนอนยื่นวิทยานิพนธ์ปริญญาโทต่อภาควิชาวิศวกรรมไฟฟ้าของ MIT ข้อเสนอของมันเรียบง่ายจนแทบน่าอาย: พีชคณิตบูล (Boolean algebra) ที่จอร์จ บูลคิดค้นขึ้นในปี 1854 เพื่ออธิบายการให้เหตุผลของมนุษย์ — พีชคณิตของจริงกับเท็จ ของ "และ" "หรือ" และ "ไม่" — ก็เป็นพีชคณิตของสวิตช์ที่ต่อแบบอนุกรมและขนานด้วยเช่นกัน สวิตช์ปิดคือ 1 สวิตช์เปิดคือ 0 สวิตช์สองตัวต่ออนุกรมคำนวณ AND สองตัวต่อขนานคำนวณ OR แชนนอนแสดงให้เห็นว่าวงจรรีเลย์ใดๆ ลดรูปได้แบบเดียวกับที่นักตรรกวิทยาลดรูปสูตร และสูตรบูลใดๆ ก็สร้างขึ้นจากรีเลย์ได้เช่นกัน ฉบับตีพิมพ์ "A Symbolic Analysis of Relay and Switching Circuits" (Transactions of the American Institute of Electrical Engineers 57, 1938: 713–723) ทำให้นักจิตวิทยาฮาวเวิร์ด การ์ดเนอร์ ในปี 1985 เรียกมันว่า "อาจเป็นวิทยานิพนธ์ปริญญาโทที่สำคัญที่สุด และมีชื่อเสียงที่สุด ของศตวรรษ" ("possibly the most important, and also the most famous, master's thesis of the century") (History of Information) ตรรกศาสตร์เป็นคำอธิบายความคิดมาสองพันปีแล้ว หลังแชนนอนมันก็กลายเป็นแผนผังการเดินสายไฟด้วย

That collision was not a straight line, and the straight line usually told is wrong. Aristotle's *Prior Analytics* had fixed logic's shape for two millennia: the syllogism, a machine of exactly three lines — "All Greeks are mortal; Socrates is a Greek; therefore Socrates is mortal" — valid by form alone, regardless of what the words meant. It was rigorous, and it was also nearly frozen; logicians spent centuries cataloguing which syllogistic figures were valid rather than extending the method. Boole's *An Investigation of the Laws of Thought* (1854) broke the freeze by algebraizing it: propositions became equations over 0 and 1, "and" became multiplication, "or" became addition, and inference became solving. This looks, in hindsight, like a rediscovery of Leibniz's seventeenth-century dream of a *calculus ratiocinator* — a calculus for settling arguments — and it is tempting to tell the story as an unbroken inheritance. It was not. Boole did not know of Leibniz's anticipations of his

algebra when *Laws of Thought* appeared; he learned of them afterward, from Robert Leslie Ellis, and was reportedly delighted rather than deflated to find Leibniz had glimpsed the same terrain two centuries earlier (SEP, *Leibniz's Influence on 19th-Century Logic*). Gottlob Frege, by contrast, did claim Leibniz's mantle deliberately when he built the next floor. The seam between the two disciplines was rediscovered at least as often as it was handed down — evidence, again, that both keep arriving at the same idea because they are excavating the same ground independently, not passing a single torch hand to hand.

> การปะทะกันนั้นไม่ใช่เส้นตรง และเส้นตรงที่มักถูกเล่ากันก็ผิด Prior Analytics ของอริสโตเติลกำหนดรูปร่างของตรรกศาสตร์ไว้สองพันปี: ซิลโลจิซึม (syllogism) เครื่องจักรสามบรรทัดเป๊ะๆ — "ชาวกรีกทุกคนตาย โสกราตีสเป็นชาวกรีก ฉะนั้น โสกราตีสตาย" — สมเหตุสมผลด้วยรูปแบบล้วนๆ ไม่ว่าถ้อยคำจะหมายถึงอะไรก็ตาม มันเข้มงวดรัดกุม และก็เกือบจะแข็งตัวด้วย นักตรรกวิทยาใช้เวลาหลายศตวรรษไล่จัดหมวดหมู่ว่ารูปแบบซิลโลจิซึมแบบไหนสมเหตุสมผล แทนที่จะขยายวิธีการออกไป An Investigation of the Laws of Thought (1854) ของบูลทำลายความแข็งตัวนั้นด้วยการแปลงให้เป็นพีชคณิต: ประพจน์กลายเป็นสมการเหนือ 0 กับ 1 "และ" กลายเป็นการคูณ "หรือ" กลายเป็นการบวก และการอนุมานกลายเป็นการแก้สมการ เมื่อมองย้อนกลับไป สิ่งนี้ดูเหมือนการค้นพบซ้ำความฝันของไลบ์นิซในศตวรรษที่สิบเจ็ดเรื่อง calculus ratiocinator — แคลคูลัสสำหรับยุติข้อโต้แย้ง — และมันชวนให้เล่าเรื่องนี้เป็นมรดกที่สืบทอดต่อกันมาไม่ขาดสาย แต่ไม่ใช่ บูลไม่รู้เรื่องที่ไลบ์นิซคาดการณ์พีชคณิตของเขาไว้ล่วงหน้าตอนที่ Laws of Thought ตีพิมพ์ เขารู้ทีหลัง จากโรเบิร์ต เลสลี เอลลิส และมีรายงานว่าเขายินดีมากกว่าจะห่อเหี่ยวใจ ที่พบว่าไลบ์นิซเคยเห็นแวบหนึ่งของดินแดนเดียวกันนี้มาก่อนสองศตวรรษ (สารานุกรมปรัชญาสแตนฟอร์ด (SEP), Leibniz's Influence on 19th-Century Logic) ในทางกลับกัน กอทท์ล็อบ เฟรเกอ้างสิทธิ์เสื้อคลุมของไลบ์นิซอย่างจงใจ เมื่อเขาสร้างชั้นถัดไป ตะเข็บระหว่างสองสาขาวิชาถูกค้นพบใหม่บ่อยพอๆ กับที่ถูกส่งต่อกันมา — เป็นหลักฐานอีกครั้งว่าทั้งคู่ยังคงมาถึงความคิดเดียวกัน เพราะทั้งคู่กำลังขุดค้นพื้นดินเดียวกันอย่างเป็นอิสระต่อกัน ไม่ใช่ส่งต่อคบเพลิงเล่มเดียวจากมือสู่มือ

Frege's *Begriffsschrift* (1879) is the second principle this chapter needs: quantification. Boole's algebra could say "some men are wise" only clumsily, as a fact about classes; it had no clean way to express "for every x, if x is a man, x is mortal" as a single manipulable structure with a variable running through it. Frege built exactly that — a notation, admittedly nothing like the ∀ and ∃ symbols used today, but expressively equivalent to them, in which generality and existence become operators binding a variable inside a formula. This is the single move that let logic swallow the reasoning used throughout ordinary mathematics, not just its syllogistic fragment — and it is also, not incidentally, the ancestor of the variable in every programming language: a slot that a quantifier, or a `for` loop, ranges over.

> Begriffsschrift (1879) ของเฟรเกคือหลักการที่สองที่บทนี้ต้องการ: การบ่งปริมาณ (quantification) พีชคณิตของบูลพูดได้แค่อย่างเงอะงะว่า "มนุษย์บางคนฉลาด" ในฐานะข้อเท็จจริงเรื่องคลาส มันไม่มีวิธีสะอาดที่จะแสดง "สำหรับทุก x ถ้า x เป็นมนุษย์ x ตาย" ให้เป็นโครงสร้างเดียวที่จัดการได้ โดยมีตัวแปรวิ่งผ่านตลอด เฟรเกสร้างสิ่งนั้นขึ้นมาพอดี — สัญกรณ์ที่ต้องยอมรับว่าไม่เหมือนสัญลักษณ์ ∀ กับ ∃ ที่ใช้กันทุกวันนี้เลย แต่เทียบเท่าในเชิงการแสดงออก ที่ซึ่งความเป็นสากลกับการมีอยู่กลายเป็นตัวดำเนินการที่ผูกตัวแปรไว้ในสูตร นี่คือการเคลื่อนไหวเดียวที่ทำให้ตรรกศาสตร์กลืนการให้เหตุผลที่ใช้ทั่วทั้งคณิตศาสตร์ทั่วไป ไม่ใช่แค่เศษเสี้ยวของซิลโลจิซึม — และมันก็เป็นบรรพบุรุษของตัวแปรในทุกภาษาโปรแกรมด้วย ไม่ใช่เรื่องบังเอิญ: ช่องว่างที่ตัวบ่งปริมาณ (quantifier) หรือลูป `for` วิ่งไล่ผ่าน

The third principle is the split Shannon's circuits do not resolve: proof is not truth. In 1929, in his doctoral dissertation under Hans Hahn, Kurt Gödel proved first-order logic *complete* — answering a question Hilbert and Ackermann had posed the year before, he showed that every logically valid formula of first-order logic has a formal proof, and (the companion property, soundness) that every provable formula is valid. Two years later, the same man showed the coin's other face: any consistent formal system rich enough to contain elementary arithmetic must be *incomplete* — it contains true arithmetical statements it cannot prove. First-order logic itself is complete; arithmetic built on top of it is not. The two results are not in tension, because they answer different questions about different systems, but to-

gether they mean something exact and permanent: mechanizing inference (soundness and completeness for logic) is not the same project as mechanizing all of mathematical truth (which Gödel showed cannot be finished). A circuit can decide whether an argument's *form* is valid. It cannot decide whether every true statement about numbers has a proof, because some do not.

> หลักการที่สามคือความแยกที่วงจรของแชนนอนไม่ได้แก้ไข: การพิสูจน์ไม่ใช่ความจริง ในปี 1929 ในวิทยานิพนธ์ปริญญาเอกภายใต้ฮันส์ ฮาห์น เคิร์ท เกอเดิลพิสูจน์ว่าตรรกศาสตร์อันดับหนึ่งสมบูรณ์ (completeness) — ตอบคำถามที่ฮิลแบร์ทกับอัคเคอร์มันน์ตั้งไว้เมื่อปีก่อนหน้า เขาแสดงให้เห็นว่าทุกสูตรที่สมเหตุสมผลเชิงตรรกะในตรรกศาสตร์อันดับหนึ่งมีบทพิสูจน์ในระบบรูปนัย (formal system) และ (คุณสมบัติคู่กันคือความสมเหตุสมผลเชิงตรรกะ (soundness)) ว่าทุกสูตรที่พิสูจน์ได้ก็สมเหตุสมผล สองปีต่อมา ชายคนเดียวกันแสดงด้านตรงข้ามของเหรียญ: ระบบรูปนัยที่สมนัยกันเองใดๆ ที่อุดมพอจะบรรจุเลขคณิตพื้นฐานไว้ได้ ต้องไม่สมบูรณ์ — มันบรรจุข้อความเลขคณิตที่เป็นจริงแต่พิสูจน์ไม่ได้ ตัวตรรกศาสตร์อันดับหนึ่งเองสมบูรณ์ แต่เลขคณิตที่สร้างขึ้นบนมันไม่สมบูรณ์ ผลลัพธ์ทั้งสองไม่ได้ขัดแย้งกัน เพราะมันตอบคำถามต่างกันเกี่ยวกับระบบต่างกัน แต่รวมกันแล้วมันหมายถึงบางสิ่งที่แน่นอนและถาวร: การทำให้การอนุมานเป็นกลไก (ความสมเหตุสมผลเชิงตรรกะและความสมบูรณ์ของตรรกศาสตร์) ไม่ใช่โครงการเดียวกับการทำให้ความจริงทางคณิตศาสตร์ทั้งหมดเป็นกลไก (ซึ่งเกอเดิลแสดงแล้วว่าทำให้เสร็จสมบูรณ์ไม่ได้) วงจรหนึ่งตัดสินได้ว่ารูปแบบของข้อโต้แย้งสมเหตุสมผลหรือไม่ แต่มันตัดสินไม่ได้ว่าทุกข้อความจริงเกี่ยวกับตัวเลขมีบทพิสูจน์หรือไม่ เพราะบางข้อความไม่มี

The fourth principle turns logic from a way of checking arguments into a way of writing programs. Robert Kowalski's 1979 slogan "Algorithm = Logic + Control" (*Communications of the* ACM 22, no. 7: 424–436) proposed splitting any procedure into a logic component — what is true, stated declaratively — and a control component — how a machine searches for it. Alain Colmerauer and Philippe Roussel had already built the proof of concept in Marseille in the summer of 1972, working out the top-down search strategy with Kowalski himself during his visits from Edinburgh: Prolog, a language

in which a program is literally a set of logical clauses, and running the program means asking the machine to prove a theorem from them by resolution (Kowalski, "The Early Years of Logic Programming"). For the first time, writing a specification and writing an executable were, formally, the same act.

> หลักการที่สี่เปลี่ยนตรรกศาสตร์จากวิธีตรวจสอบข้อโต้แย้งให้กลายเป็นวิธีเขียนโปรแกรม สโลแกนปี 1979 ของโรเบิร์ต โควาลสกี "Algorithm = Logic + Control" (Communications of the ACM 22, no. 7: 424–436) เสนอให้แยกกระบวนวิธีใดๆ ออกเป็นส่วนตรรกะ — สิ่งที่เป็นจริง ระบุแบบประกาศ (declarative) — กับส่วนควบคุม — เครื่องจักรค้นหามันอย่างไร อาแล็ง กอลเมโรกับฟีลิป รูแซลสร้างการพิสูจน์แนวคิดนี้ไว้แล้วที่มาร์แซย์ในฤดูร้อนปี 1972 คิดค้นกลยุทธ์การค้นหาแบบบนลงล่างร่วมกับโควาลสกีเองระหว่างที่เขาไปเยือนจากเอดินบะระ: Prolog ภาษาที่โปรแกรมหนึ่งคือชุดอนุประโยคเชิงตรรกะโดยแท้จริง และการรันโปรแกรมหมายถึงการขอให้เครื่องพิสูจน์ทฤษฎีบทจากอนุประโยคเหล่านั้นด้วย resolution (Kowalski, "The Early Years of Logic Programming") เป็นครั้งแรกที่การเขียนข้อกำหนดและการเขียนโปรแกรมที่รันได้ กลายเป็นการกระทำเดียวกันในทางรูปนัย

The fifth principle is the deepest handshake between the two courts in this entire book: the Curry–Howard correspondence. In 1934, Haskell Curry noticed that the types assignable to the combinators of combinatory logic matched, term for term, the axioms of intuitionistic implicational logic ("Functionality in Combinatory Logic," *PNAS* 20). William Howard extended the observation to full natural deduction and the lambda calculus in 1969, though the paper that made it famous, "The Formulae-as-Types Notion of Construction," was not published until 1980, in a festschrift for Curry. The correspondence says something stronger than analogy: a proposition *is* a type, a proof of that proposition *is* a program of that type, and simplifying a proof — removing a detour — *is* running the program one step. Logic and computation are not two things that resemble each other; under Curry–Howard, for a wide and growing class of systems, they are the same thing

described twice. Modern proof assistants (Coq, Agda, Lean) are, quite literally, programming languages in which a well-typed program is a certificate that a theorem is true.

> หลักการที่ห้าคือการจับมือที่ลึกที่สุดระหว่างสองศาลตลอดทั้งเล่มนี้: ความสอดคล้อง Curry–Howard ในปี 1934 แฮสเคลล์ เคอร์รีสังเกตเห็นว่าชนิด (type) ที่กำหนดให้กับตัวรวมของตรรกะเชิงการจัด (combinatory logic) นั้นตรงกัน ทีละพจน์ กับสัจพจน์ของตรรกะสัญชานนิยม (intuitionistic logic) เชิงเงื่อนไข ("Functionality in Combinatory Logic," PNAS 20) วิลเลียม ฮาวเวิร์ดขยายข้อสังเกตนี้ไปถึงการนิรนัยธรรมชาติ (natural deduction) เต็มรูปแบบและแคลคูลัสแลมบ์ดา (lambda calculus) ในปี 1969 แม้ว่าบทความที่ทำให้แนวคิดนี้มีชื่อเสียง "The Formulae-as-Types Notion of Construction" จะไม่ได้ตีพิมพ์จนถึงปี 1980 ในหนังสือรวมบทความเชิดชูเกียรติเคอร์รี ความสอดคล้องนี้บอกอะไรที่แข็งแรงกว่าการเปรียบเทียบ: ประพจน์หนึ่งคือชนิดหนึ่ง บทพิสูจน์ของประพจน์นั้นคือโปรแกรมของชนิดนั้น และการลดรูปบทพิสูจน์ — การตัดทางอ้อมทิ้ง — คือการรันโปรแกรมไปหนึ่งขั้น ตรรกศาสตร์กับการคำนวณไม่ใช่สองสิ่งที่คล้ายกัน ภายใต้ Curry–Howard สำหรับกลุ่มระบบที่กว้างขวางและกำลังขยายตัว มันคือสิ่งเดียวกันที่ถูกอธิบายสองครั้ง proof assistant (ตัวช่วยพิสูจน์) สมัยใหม่ (Coq, Agda, Lean) เป็นภาษาโปรแกรมโดยแท้จริง ที่ซึ่งโปรแกรมที่พิมพ์ถูกต้อง (well-typed) คือใบรับรองว่าทฤษฎีบทหนึ่งเป็นจริง

The people, briefly. Boole was self-taught, working as a schoolmaster in Lincoln before a professorship at Cork; Frege spent most of his career unread, his logic noticed chiefly because Bertrand Russell wrote to him in 1902 to point out a contradiction in it. Kurt Gödel worked in near isolation in Vienna and later Princeton, distrustful to the point of starving himself in his final year. Haskell Curry and William Howard are barely names outside logic

departments — fitting, for a correspondence whose own publication was delayed eleven years, as if the ideas in this chapter have a habit of arriving before anyone is ready to print them.

> **ผู้คน โดยย่อ** บูลเรียนรู้ด้วยตัวเอง ทำงานเป็นครูใหญ่ในลินคอล์นก่อนได้เป็นศาสตราจารย์ที่คอร์ก เฟรเกใช้ชีวิตการทำงานส่วนใหญ่โดยไม่มีใครอ่านผลงานตรรกะของเขาได้รับความสนใจก็เพราะเบอร์ทรันด์ รัสเซลล์เขียนจดหมายถึงเขาในปี 1902 เพื่อชี้ข้อขัดแย้งในนั้น เคิร์ท เกอเดิลทำงานอย่างโดดเดี่ยวเกือบสิ้นเชิงในเวียนนาและภายหลังที่พรินซ์ตัน ระแวงจนถึงขั้นอดอาหารตัวเองตายในปีสุดท้ายของชีวิต แฮสเคลล์ เคอร์รีกับวิลเลียม ฮาวเวิร์ดแทบไม่เป็นที่รู้จักนอกภาควิชาตรรกศาสตร์ — เหมาะสมดี สำหรับความสอดคล้องที่การตีพิมพ์ของมันเองล่าช้าไปสิบเอ็ดปี ราวกับว่าความคิดในบทนี้มีนิสัยมาถึงก่อนที่ใครจะพร้อมตีพิมพ์มัน

Connections · เชื่อมโยง

Chapter 00 is this chapter's parent: the Frege–Russell–Hilbert–Gödel–Turing pipeline is narrated there in full; this chapter stays inside logic's own family quarrel. Chapter 09 returns to 1936 to ask what the Church–Turing thesis actually claims. Chapter 11 owns Shannon's *later* severing of information from meaning in 1948; this chapter's Shannon stops at 1937, where circuits meet Boole and nothing yet has been said about meaning. Chapter 12 takes Curry–Howard to its full depth, through Brouwer's intuitionism and mechanized mathematics.

> บทที่ 00 คือบทแม่ของบทนี้: สายพานเฟรเก–รัสเซลล์–ฮิลแบร์ท–เกอเดิล–ทัวริงถูกเล่าไว้ครบถ้วนที่นั่น บทนี้อยู่แค่ในเรื่องทะเลาะกันเองในครอบครัวตรรกศาสตร์ บทที่ 09 ย้อนกลับไปปี 1936 เพื่อถามว่าข้อตั้ง Church–Turing อ้างอะไรกันแน่ บทที่ 11 เป็นเจ้าของเรื่องการตัดความหมายออกจากสารสนเทศของแชนนอนในภายหลัง ปี 1948 ส่วนแชนนอนของบทนี้หยุดอยู่แค่ปี 1937 ที่ซึ่งวงจรมาบรรจบกับบูล และยังไม่มีการพูดถึงความหมายเลย บทที่ 12 พาความสอดคล้อง Curry–Howard ไปถึงความลึกเต็มรูปแบบ ผ่านตรรกะสัญชานิยมของบราวเวอร์และคณิตศาสตร์เชิงกลไก

Open questions · คำถามที่ยังเปิดอยู่

If Boole and Leibniz, and later Frege, kept independently finding the same seam, is a mathematized logic something humanity was always going to discover once algebra and doubt about the syllogism coincided — or was the coincidence itself contingent? And if proof and program are truly the same object under Curry–Howard, which discipline should be said to have discovered the other: did computing find out it was secretly doing logic, or did logic find out it was secretly executable?

> ถ้าบูลกับไลบ์นิซ และภายหลังเฟรเก ต่างค้นพบตะเข็บเดียวกันโดยอิสระต่อกันซ้ำแล้วซ้ำเล่า ตรรกศาสตร์ที่แปลงเป็นคณิตศาสตร์นั้นเป็นสิ่งที่มนุษยชาติจะค้นพบอยู่ดีไม่ว่าอย่างไร เมื่อพีชคณิตกับความสงสัยในซิลโลจิซึมมาบรรจบกันพอดี — หรือความบังเอิญนั้นเองก็เป็นเรื่องไม่แน่นอน และถ้าบทพิสูจน์กับโปรแกรมเป็นวัตถุเดียวกันจริงๆ ภายใต้ Curry–Howard สาขาไหนควรถูกกล่าวว่าเป็นฝ่ายค้นพบอีกฝ่าย: การคำนวณค้นพบว่าตัวเองแอบทำตรรกศาสตร์อยู่ หรือตรรกศาสตร์ค้นพบว่าตัวเองแอบรันได้อยู่กันแน่

Go deeper · อ่านต่อ

- Ernest Nagel and James R. Newman, *Gödel's Proof*, New York University Press, 1958 — still the clearest short account of what completeness and incompleteness do and do not say.

  > Ernest Nagel and James R. Newman, Gödel's Proof, New York University Press, 1958 — ยังคงเป็นคำอธิบายสั้นที่ชัดเจนที่สุด ว่าความสมบูรณ์กับความไม่สมบูรณ์บอกอะไรและไม่บอกอะไร

- Claude Shannon, "A Symbolic Analysis of Relay and Switching Circuits," *Transactions of the American Institute of Electrical Engineers* 57 (1938): 713–723; William Howard, "The Formulae-as-Types Notion of Construction," in *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, eds. J.P. Seldin and J.R. Hindley, Academic Press, 1980 (written 1969).

> Claude Shannon, "A Symbolic Analysis of Relay and Switching Circuits," Transactions of the American Institute of Electrical Engineers 57 (1938): 713–723; William Howard, "The Formulae-as-Types Notion of Construction," in To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, eds. J.P. Seldin and J.R. Hindley, Academic Press, 1980 (เขียนขึ้นปี 1969)

— *Gödel's Incompleteness Theorems*, Stanford Encyclopedia of Philosophy.

> Gödel's Incompleteness Theorems, SEP

CHAPTER 02

Metaphysics

อภิปรัชญา

Part I's second hearing: with logic's family quarrel behind it, the court turns to a plainer question — what kind of object is sitting in the dock at all.

การไต่สวนครั้งที่สองของภาค I: เมื่อเรื่องทะเลาะกันเองในครอบครัวของตรรกศาสตร์ผ่านพ้นไปแล้ว ศาลหันมาสู่คำถามที่ตรงไปตรงมากว่า — สิ่งชนิดไหนกันแน่ที่นั่งอยู่ในคอกจำเลย

What kind of thing is a program?

โปรแกรมคือสิ่งชนิดไหนกันแน่

In 1988 Hilary Putnam published *Representation and Reality* (MIT Press) with what he called a theorem: every ordinary open physical system — anything not perfectly sealed off from its environment — implements every finite-state automaton. Two years later, in his presidential address to the American Philosophical Association, John Searle pushed the same point to a memorable extreme: "the wall behind my back is right now implementing the Wordstar program" — because some pattern in its molecular motion is isomorphic with Wordstar's formal structure ("Is the Brain a Digital Computer?", *Proceedings and Addresses of the* APA 64, 1990: 21–37). If a wall runs Microsoft's word processor, then computation is not a fact about the world — it is a story an observer chooses to tell about any sufficiently complicated system, brains included. Putnam's own triviality result, aimed originally at his own earlier theory, exploded computational functionalism's central

promise: that being a mind is a matter of running the right program, no matter what you're made of. If any wall runs any program, running the right program cannot be what makes something a mind.

> ในปี 1988 ฮิลารี พัทนัมตีพิมพ์ Representation and Reality (MIT Press) พร้อมสิ่งที่เขาเรียกว่าทฤษฎีบทหนึ่ง: ระบบกายภาพเปิดทั่วไปทุกระบบ — อะไรก็ตามที่ไม่ได้ถูกปิดกั้นจากสิ่งแวดล้อมของมันอย่างสมบูรณ์ — นำไปใช้จริง (implementation) ออโตมาตาสถานะจำกัด (finite-state automaton) ทุกตัว สองปีต่อมา ในปาฐกถาประธานาธิบดีต่อสมาคมปรัชญาอเมริกัน จอห์น เซิร์ลผลักประเด็นเดียวกันไปสู่จุดสุดโต่งที่น่าจดจำ: "กำแพงข้างหลังผมกำลังนำโปรแกรม Wordstar ไปใช้จริงอยู่ตอนนี้" — เพราะแบบรูปหนึ่งในการเคลื่อนไหวระดับโมเลกุลของมันมีสัณฐานเดียวกัน (isomorphic) กับโครงสร้างรูปนัยของ Wordstar ("Is the Brain a Digital Computer?", Proceedings and Addresses of the APA 64, 1990: 21–37) ถ้ากำแพงหนึ่งรันโปรแกรมประมวลผลคำของไมโครซอฟท์ได้ การคำนวณก็ไม่ใช่ข้อเท็จจริงเกี่ยวกับโลก — มันคือเรื่องเล่าที่ผู้สังเกตเลือกเล่าเกี่ยวกับระบบใดๆ ที่ซับซ้อนมากพอ รวมถึงสมองด้วย ผลลัพธ์เรื่องความจิ๊บจ๊อย (triviality argument) ของพัทนัมเอง ซึ่งเดิมทีมุ่งเป้าไปที่ทฤษฎีก่อนหน้าของตัวเขาเอง ทำลายคำมั่นสัญญาหลักของหน้าที่นิยมเชิงคำนวณ (computational functionalism) จนแหลกสลาย: ที่ว่าการเป็นจิตหนึ่งคือเรื่องของการรันโปรแกรมที่ถูกต้อง ไม่ว่าคุณจะทำจากอะไรก็ตาม ถ้ากำแพงใดๆ รันโปรแกรมใดๆ ก็ได้ การรันโปรแกรมที่ถูกต้องก็ไม่อาจเป็นสิ่งที่ทำให้บางสิ่งกลายเป็นจิตได้

The wall argument is where this chapter has to start, because it forces the question philosophy usually skips: before asking whether a program can think, ask what a program even *is*.

> ข้อโต้แย้งเรื่องกำแพงคือจุดที่บทนี้ต้องเริ่มต้น เพราะมันบังคับให้เผชิญคำถามที่ปรัชญามักข้ามไป: ก่อนจะถามว่าโปรแกรมคิดได้หรือไม่ ต้องถามก่อนว่าโปรแกรมคืออะไรกันแน่

The first principle is the split between the abstract and the concrete — computing's own version of a distinction Charles Sanders Peirce drew for language in 1906: type versus token. Count the letters on this page and you

can answer two different, equally correct questions — how many letter-shapes appear (tokens), and how many distinct letters of the alphabet they instantiate (types) ("Prolegomena to an Apology for Pragmaticism," *The Monist* 16, no. 4, 1906: 505–506). A piece of software works the same way. "Microsoft Word" the program is a type — an abstract pattern of instructions, nowhere in particular, no more locatable than the number seven. The install on your laptop, and the running process consuming memory on it right now, are tokens — concrete, datable, capable of crashing. When your copy of Word freezes, Word itself, the abstract program, has not broken; a token has. Software's whole industry runs on treating the type as the real product and the token as a disposable instance of it — which is exactly why a "critical vulnerability in Word" can be announced before your particular copy has even loaded.

> หลักการแรกคือความแยกระหว่างสิ่งนามธรรมกับสิ่งรูปธรรม — ฉบับของวงการคำนวณเองสำหรับความแตกต่างที่ชาลส์ แซนเดอร์ส เพิร์ซวางไว้สำหรับภาษาในปี 1906: ชนิด (type) กับตัวอย่างจริง (token) นับตัวอักษรบนหน้านี้ แล้วคุณจะตอบคำถามสองข้อที่ต่างกันแต่ถูกต้องเท่ากันได้ — มีรูปตัวอักษรปรากฏกี่ตัว (ตัวอย่างจริง) และมันเป็นตัวแทนของตัวอักษรที่แตกต่างกันในอักษรกี่ตัว (ชนิด) ("Prolegomena to an Apology for Pragmaticism," The Monist 16, no. 4, 1906: 505–506) ซอฟต์แวร์ชิ้นหนึ่งทำงานแบบเดียวกัน "Microsoft Word" ในฐานะโปรแกรมคือชนิดหนึ่ง — แบบรูปนามธรรมของคำสั่ง ไม่ได้อยู่ที่ไหนโดยเฉพาะ ระบุตำแหน่งไม่ได้มากไปกว่าเลขเจ็ด สิ่งที่ติดตั้งอยู่บนแล็ปท็อปของคุณ และโพรเซสที่กำลังรันกินหน่วยความจำอยู่ตอนนี้ คือตัวอย่างจริง — เป็นรูปธรรม ระบุวันที่ได้ และล่มได้ เมื่อสำเนา Word ของคุณค้าง ตัว Word เอง โปรแกรมนามธรรมนั้น ไม่ได้เสีย มีแค่ตัวอย่างจริงตัวหนึ่งที่เสีย อุตสาหกรรมซอฟต์แวร์ทั้งหมดตั้งอยู่บนการปฏิบัติต่อชนิดเสมือนเป็นสินค้าจริง และตัวอย่างจริงเป็นแค่กรณีใช้แล้วทิ้งของมัน — ซึ่งเป็นเหตุผลเป๊ะๆ ว่าทำไม "ช่องโหว่ร้ายแรงใน Word" ถึงถูกประกาศได้ ก่อนที่สำเนาของคุณโดยเฉพาะจะโหลดขึ้นมาด้วยซ้ำ

The second principle is identity, and it is harder than it sounds: when are two programs the *same* program? Take a bubble sort written in Python and the identical algorithm rewritten, variable for variable, in C. Same steps,

same inputs and outputs, different languages — same program, or two? Now take one Python file compiled by two different optimizing compilers into two binaries that never produce a single identical machine instruction but always agree on output — same program? Philosophers of computation have no settled answer, and the disagreement tracks a real fork: identify a program with its *behavior* (the input–output function it computes) and the two binaries are one program; identify it with its *structure* (the algorithm, the specific steps) and a compiler's optimizations may quietly turn it into something else that merely agrees on outputs so far tested. Software engineers live this dispute daily without naming it, every time they ask whether a refactor "changed the program" or just changed its clothes.

> หลักการที่สองคือเรื่องอัตลักษณ์ (identity) และมันยากกว่าที่ฟังดู: โปรแกรมสองตัวเป็นโปรแกรมเดียวกันเมื่อไหร่ ลองนึกถึง bubble sort ที่เขียนด้วยไพธอน กับอัลกอริทึมเดียวกันเป๊ะที่เขียนใหม่ ตัวแปรต่อตัวแปร ด้วยภาษา C ขั้นตอนเดียวกัน อินพุตเอาต์พุตเดียวกัน ภาษาต่างกัน — โปรแกรมเดียวกัน หรือสองโปรแกรม ทีนี้ลองนึกถึงไฟล์ไพธอนไฟล์เดียวที่ถูกคอมไพล์ด้วยคอมไพเลอร์เพิ่มประสิทธิภาพสองตัวที่ต่างกัน ออกมาเป็นไบนารีสองไฟล์ที่ไม่เคยผลิตคำสั่งเครื่องที่เหมือนกันเป๊ะสักคำสั่งเดียว แต่ให้ผลลัพธ์ตรงกันเสมอ — โปรแกรมเดียวกันหรือไม่ นักปรัชญาแห่งการคำนวณไม่มีคำตอบที่ตกลงกันได้ และความเห็นต่างนี้ตามรอยทางแยกจริง: ระบุโปรแกรมด้วยพฤติกรรมของมัน (ฟังก์ชันอินพุต-เอาต์พุตที่มันคำนวณ) แล้วไบนารีสองไฟล์นั้นก็เป็นโปรแกรมเดียว ระบุมันด้วยโครงสร้างของมัน (อัลกอริทึม ขั้นตอนเฉพาะ) แล้วการเพิ่มประสิทธิภาพของคอมไพเลอร์อาจแอบเปลี่ยนมันให้กลายเป็นอะไรอีกอย่างที่แค่บังเอิญให้ผลลัพธ์ตรงกันเท่าที่เคยทดสอบมา วิศวกรซอฟต์แวร์ใช้ชีวิตอยู่กับข้อพิพาทนี้ทุกวันโดยไม่เคยเรียกชื่อมัน ทุกครั้งที่พวกเขาถามว่าการ refactor "เปลี่ยนโปรแกรม" ไปแล้วหรือแค่เปลี่ยนเสื้อผ้าให้มัน

The third principle is the wall itself, formalized. Putnam's proof works by constructing, after the fact, a mapping from a chosen sequence of the wall's molecular microstates onto the automaton's state transitions — a mapping guaranteed to exist for any finite stretch of time because there are always enough distinguishable microstates to go around. David Chalmers's reply, "Does a Rock Implement Every Finite-State Automaton?" (*Synthese* 108,

1996: 309–333), grants the construction and denies it proves what Putnam needs. A genuine implementation, Chalmers argues, must track the automaton's transitions not just along the one trajectory the world happened to take, but *counterfactually* — for every input the automaton could have received, the physical system's causal structure must have produced the matching state, whether or not that input actually occurred. Putnam's wall satisfies the mapping only along the single path molecules actually followed; feed the wall a different "input" and nothing in it is disposed to track the automaton's corresponding response, because nothing in a wall is wired to respond to inputs at all. A laptop running WordStar, by contrast, really is built so that pressing a different key really would produce the counterfactually correct state. The reply does not make implementation cheap; it makes it a genuine, checkable, causal fact about a system's dispositions — restoring exactly the distinction Searle's wall was built to erase.

> หลักการที่สามคือตัวข้อโต้แย้งเรื่องกำแพงเอง ที่ถูกทำให้เป็นรูปนัย บทพิสูจน์ของพัทนัมทำงานด้วยการสร้าง ภายหลังเหตุการณ์ การจับคู่จากลำดับที่เลือกไว้ของสถานะจุลภาคระดับโมเลกุลของกำแพง ไปยังการเปลี่ยนสถานะของออโตมาตา — การจับคู่ที่รับประกันว่ามีอยู่จริงสำหรับช่วงเวลาจำกัดใดๆ เพราะมีสถานะจุลภาคที่แยกแยะได้เพียงพอเสมอ คำตอบโต้ของเดวิด ชาลเมอร์ส "Does a Rock Implement Every Finite-State Automaton?" (Synthese 108, 1996: 309–333) ยอมรับการสร้างนี้แต่ปฏิเสธว่ามันพิสูจน์สิ่งที่พัทนัมต้องการ ชาลเมอร์สโต้แย้งว่าการนำไปใช้จริงที่แท้จริงต้องติดตามการเปลี่ยนสถานะของออโตมาตาไม่ใช่แค่ตามวิถีทางเดียวที่โลกบังเอิญเดินไป แต่ต้องเป็นแบบ counterfactual (สมมติสวนความจริง) ด้วย — สำหรับทุกอินพุตที่ออโตมาตาอาจได้รับ โครงสร้างเชิงเหตุปัจจัยของระบบกายภาพต้องผลิตสถานะที่ตรงกันออกมา ไม่ว่าอินพุตนั้นจะเกิดขึ้นจริงหรือไม่ กำแพงของพัทนัมสอดคล้องกับการจับคู่นั้นแค่ตามเส้นทางเดียวที่โมเลกุลเดินไปจริง ป้อน "อินพุต" อื่นให้กำแพง ไม่มีสิ่งใดในนั้นพร้อมจะติดตามการตอบสนองที่สอดคล้องของออโตมาตา เพราะไม่มีสิ่งใดในกำแพงถูกต่อสายไว้ให้ตอบสนองต่ออินพุตเลย แล็ปท็อปที่รัน WordStar ในทางตรงข้าม ถูกสร้างขึ้นจริงๆ ให้การกดปุ่มอื่นจะผลิตสถานะที่ถูกต้องแบบ counterfactual ได้จริง คำตอบโต้นี้ไม่ได้ทำให้การ

> นำไปใช้จริงเป็นเรื่องหาได้ง่าย แต่ทำให้มันเป็นข้อเท็จจริงเชิงเหตุปัจจัยที่แท้จริงและตรวจสอบได้เกี่ยวกับนิสัยโน้มเอียงของระบบหนึ่ง — ซึ่งฟื้นคืนความแตกต่างที่กำแพงของเซิร์ลถูกสร้างขึ้นมาเพื่อลบทิ้งพอดี

The fourth principle takes the same question into worlds built entirely of information: virtual realism. In *Reality+: Virtual Worlds and the Problems of Philosophy* (W. W. Norton, 2022) and the paper that preceded it, "The Virtual and the Real" (*Disputatio* 9, no. 46, 2017: 309–352), Chalmers argues that objects inside a simulated world are not lesser or fictional — they are real digital objects, meeting the marks philosophers use to certify anything as real: they exist, they have causal powers, they are not merely someone's say-so, and they are mind-independent once the simulation is running. A sword forged inside a game world is not made of steel, but it is made of data structures that really exist, really can be transferred, and really can be sold for currency that buys real coffee. Chalmers's target is not that virtual and physical objects are identical in kind — a virtual river is not H2O — but that being virtual is not the same as being unreal; it is one more way of being real, alongside being physical.

> หลักการที่สี่พาคำถามเดียวกันนี้เข้าไปในโลกที่สร้างขึ้นจากสารสนเทศล้วนๆ: สัจนิยมเสมือน (virtual realism) ใน Reality+: Virtual Worlds and the Problems of Philosophy (W. W. Norton, 2022) และบทความก่อนหน้ามัน "The Virtual and the Real" (Disputatio 9, no. 46, 2017: 309–352) ชาลเมอร์สโต้แย้งว่าวัตถุภายในโลกจำลองไม่ได้ด้อยกว่าหรือเป็นเรื่องแต่ง — มันคือวัตถุดิจิทัลจริง ที่เข้าเกณฑ์ที่นักปรัชญาใช้รับรองว่าสิ่งใดสิ่งหนึ่งเป็นจริง: มันมีอยู่ มันมีอำนาจเชิงเหตุปัจจัย มันไม่ใช่แค่คำพูดของใครคนหนึ่ง และมันเป็นอิสระจากจิตเมื่อการจำลองกำลังทำงานอยู่ ดาบที่ตีขึ้นภายในโลกเกมไม่ได้ทำจากเหล็กกล้า แต่มันทำจากโครงสร้างข้อมูลที่มีอยู่จริง โอนย้ายได้จริง และขายแลกเป็นสกุลเงินที่ซื้อกาแฟจริงได้จริง เป้าหมายของชาลเมอร์สไม่ใช่ว่าวัตถุเสมือนกับวัตถุกายภาพเหมือนกันโดยชนิด — แม่น้ำเสมือนไม่ใช่ H2O — แต่คือการเป็นสิ่งเสมือนไม่เหมือนกับการไม่จริง มันคืออีกวิธีหนึ่งของการเป็นจริง ควบคู่ไปกับการเป็นกายภาพ

The fifth principle is a horizon this chapter only touches: the ontology of information itself. Luciano Floridi's *informational structural realism* (*The Philosophy of Information*, Oxford University Press, 2011) proposes that what ultimately exists is not substance but structured information — "dedomena," primordial data-like differences — of which physical objects and virtual objects alike are patterns at different levels of abstraction. If Floridi is right, the type/token puzzle and the virtual-object puzzle are not two separate metaphysical headaches produced by computers; they are two symptoms of the same deeper claim, that information was always the more basic category and substance the derived one. Chapter 11 audits that claim on its own ground, against Shannon's severing of information from meaning; this chapter only notes that metaphysics, once it starts asking what a program is, cannot easily stop there.

> หลักการที่ห้าคือขอบฟ้าที่บทนี้แตะเพียงผิวเผิน: ภววิทยา (ontology) ของสารสนเทศเอง สัจนิยมเชิงโครงสร้างสารสนเทศ (informational structural realism) ของลูชาโน ฟลอริดี (The Philosophy of Information, Oxford University Press, 2011) เสนอว่าสิ่งที่มีอยู่จริงในท้ายที่สุดไม่ใช่สสาร แต่คือสารสนเทศที่มีโครงสร้าง — "dedomena" ความแตกต่างแบบข้อมูลดั้งเดิม — ซึ่งทั้งวัตถุกายภาพและวัตถุเสมือนต่างก็เป็นแบบรูปที่ระดับ abstraction ต่างกัน ถ้าฟลอริดีพูดถูก ปริศนาชนิด/ตัวอย่างจริงกับปริศนาวัตถุเสมือนก็ไม่ใช่ปัญหาปวดหัวเชิงอภิปรัชญาสองเรื่องที่แยกจากกันซึ่งคอมพิวเตอร์สร้างขึ้น แต่เป็นอาการสองอย่างของข้อกล่าวอ้างที่ลึกกว่าเดียวกัน คือสารสนเทศเป็นหมวดหมู่พื้นฐานกว่าเสมอมา และสสารคือสิ่งที่สืบเนื่องมาจากมัน บทที่ 11 ตรวจสอบข้อกล่าวอ้างนั้นบนพื้นที่ของมันเอง เทียบกับการตัดสารสนเทศออกจากความหมายของแชนนอน บทนี้เพียงตั้งข้อสังเกตว่าอภิปรัชญา เมื่อเริ่มถามว่าโปรแกรมคืออะไร ก็ไม่อาจหยุดอยู่แค่นั้นได้ง่ายๆ

The people, briefly. Hilary Putnam changed his mind about the philosophy of mind more publicly than almost anyone in the field, ending his life doubting the very functionalism he had founded. Searle's wall is a philosopher's weapon turned, characteristically, against a friendly-adjacent target — computational theories of mind he otherwise also opposes on different grounds

(see chapter 04). Chalmers has spent a career doing what this chapter shows twice: taking a claim meant to dissolve a distinction and finding the exact repair that saves it.

> **ผู้คน โดยย่อ** ฮิลารี พัทนัมเปลี่ยนใจเรื่องปรัชญาแห่งจิตอย่างเปิดเผยต่อสาธารณะมากกว่าใครแทบทุกคนในสาขานี้ จบชีวิตด้วยความสงสัยในตัวหน้าที่นิยม (functionalism) ที่เขาเองเป็นผู้ก่อตั้งขึ้นมา กำแพงของเซิร์ลคืออาวุธของนักปรัชญาที่หันไปเล็งเป้าหมายที่เป็นมิตรใกล้เคียงกัน อย่างที่เขามักทำ — ทฤษฎีจิตเชิงคำนวณ (computational theory of mind) ที่เขาเองก็คัดค้านด้วยเหตุผลอื่นเช่นกัน (ดูบทที่ 04) ชาลเมอร์สใช้อาชีพทั้งชีวิตทำสิ่งที่บทนี้แสดงให้เห็นสองครั้ง: หยิบข้อกล่าวอ้างที่ตั้งใจจะละลายความแตกต่างหนึ่ง แล้วหาทางซ่อมแซมที่แม่นยำที่ช่วยรักษามันไว้

Connections · เชื่อมโยง

Chapter 04 needs this chapter's implementation problem directly: if implementing a program is not enough to *be* a mind, functionalism about mental states inherits the same objection Searle raised here. Chapter 11 owns the full audit of information as a candidate ultimate substance. Chapter 15 asks the largest version of the virtual-realism question: is the physical universe itself the simulation.

> บทที่ 04 ต้องการปัญหาการนำไปใช้จริงของบทนี้โดยตรง: ถ้าการนำโปรแกรมไปใช้จริงไม่พอจะเป็นจิตหนึ่ง หน้าที่นิยมเรื่องสถานะจิตก็รับช่วงข้อคัดค้านเดียวกันที่เซิร์ลยกขึ้นมาที่นี่ บทที่ 11 เป็นเจ้าของการตรวจสอบเต็มรูปแบบว่าสารสนเทศเป็นผู้ท้าชิงตำแหน่งสสารสูงสุดหรือไม่ บทที่ 15 ถามคำถามเรื่องสัจนิยมเสมือนในเวอร์ชันใหญ่ที่สุด: เอกภพกายภาพเองคือการจำลองหรือไม่

Open questions · คำถามที่ยังเปิดอยู่

If Chalmers is right that implementation requires tracking counterfactual structure, is that structure an objective fact about a physical system, or does specifying "the right counterfactuals" still require an observer to say which inputs count? And can one consistently hold, as Chalmers does, that

game-world objects are real without holding that the base universe is itself a simulation — or does virtual realism quietly commit you to taking chapter 15's simulation argument more seriously than you intended?

> ถ้าชาลเมอร์สพูดถูกว่าการนำไปใช้จริงต้องอาศัยการติดตามโครงสร้างแบบ counterfactual โครงสร้างนั้นเป็นข้อเท็จจริงเชิงภาวะวิสัยเกี่ยวกับระบบกายภาพหนึ่งหรือไม่ หรือการระบุ "counterfactual ที่ถูกต้อง" ยังต้องอาศัยผู้สังเกตมาบอกว่าอินพุตไหนนับอยู่ดี และคนเราจะยึดถืออย่างสอดคล้องกันได้หรือไม่ อย่างที่ชาลเมอร์สทำ ว่าวัตถุในโลกเกมเป็นจริง โดยไม่ต้องยึดถือว่าเอกภพฐานเองก็เป็นการจำลอง — หรือสัจนิยมเสมือนแอบผูกมัดให้คุณต้องเอาข้อโต้แย้งเรื่องการจำลองของบทที่ 15 ไปคิดจริงจังกว่าที่คุณตั้งใจไว้

Go deeper · อ่านต่อ

- David J. Chalmers, *Reality+: Virtual Worlds and the Problems of Philosophy*, W. W. Norton, 2022 — the book-length case for virtual realism, built from the implementation problem up.

  > David J. Chalmers, Reality+: Virtual Worlds and the Problems of Philosophy, W. W. Norton, 2022 — ข้อเสนอระดับหนังสือทั้งเล่มสำหรับสัจนิยมเสมือน สร้างขึ้นจากปัญหาการนำไปใช้จริงเป็นฐาน

- Hilary Putnam, *Representation and Reality*, MIT Press, 1988 (the triviality argument); David J. Chalmers, "Does a Rock Implement Every Finite-State Automaton?", *Synthese* 108 (1996): 309–333 (the reply).

  > Hilary Putnam, Representation and Reality, MIT Press, 1988 (ข้อโต้แย้งเรื่องความจิ๊บจ๊อย); David J. Chalmers, "Does a Rock Implement Every Finite-State Automaton?", Synthese 108 (1996): 309–333 (คำตอบโต้)

- *Types and Tokens*, Stanford Encyclopedia of Philosophy.

  > Types and Tokens, สารานุกรมปรัชญาสแตนฟอร์ด (SEP)

CHAPTER 03

Epistemology

ญาณวิทยา

Part I's third hearing: having asked what a program is, the court now asks what it is entitled to claim it knows.

การไต่สวนครั้งที่สามของภาค I: เมื่อถามไปแล้วว่าโปรแกรมคืออะไร ศาลจึงถามต่อว่ามันมีสิทธิ์อ้างว่ารู้อะไรได้บ้าง

Does a machine's answer count as knowledge?

คำตอบของเครื่องจักรนับเป็นความรู้หรือไม่

In 1976, Kenneth Appel and Wolfgang Haken, at the University of Illinois at Urbana–Champaign, announced a proof of the four-color theorem — the claim, open since 1852, that any map can be colored with four colors so that no two adjacent regions share one. Their method reduced the infinite variety of possible maps to an unavoidable set of 1,476 configurations, each of which had to be checked for "reducibility" by machine; the computer ran roughly 1,200 hours (Robertson, Sanders, Seymour & Thomas, "The Four Color Theorem"). No mathematician has ever read that check line by line, and none ever will; the case-work is too vast, and no journal referee re-derives it. The result was the first proof of a major theorem that essentially required a computer, and it unsettled mathematicians precisely because it broke an assumption so old it had never needed stating: that a proof is something a trained human being can, in principle, follow from first line to last and certify with their own understanding. The four-color theorem is

almost certainly true. Whether the Appel–Haken argument counts as a *proof* in the older sense — something surveyable, something a mind can hold — is a question the mathematical community has never fully closed.

> ในปี 1976 เคนเนธ แอปเพลกับวุลฟ์กัง ฮาเคน ที่มหาวิทยาลัยอิลลินอยส์ เออร์บานา-แชมเปญ ประกาศบทพิสูจน์ทฤษฎีบทสี่สี — ข้อกล่าวอ้างที่เปิดค้างมาตั้งแต่ปี 1852 ว่าแผนที่ใดๆ ระบายสีได้ด้วยสี่สีโดยไม่มีสองพื้นที่ที่ติดกันใช้สีเดียวกัน วิธีการของพวกเขาลดความหลากหลายไม่จำกัดของแผนที่ที่เป็นไปได้ลงเหลือชุดที่หลีกเลี่ยงไม่ได้ของ 1,476 โครงแบบ แต่ละโครงแบบต้องถูกตรวจสอบ "การลดรูปได้" ด้วยเครื่องคอมพิวเตอร์รันอยู่ราว 1,200 ชั่วโมง (Robertson, Sanders, Seymour & Thomas, "The Four Color Theorem") ไม่มีนักคณิตศาสตร์คนใดเคยอ่านการตรวจสอบนั้นทีละบรรทัด และจะไม่มีใครทำเลย งานตรวจกรณีนั้นใหญ่เกินไป และไม่มีผู้ตรวจทานวารสารคนไหนอนุมานซ้ำมันได้ ผลลัพธ์นี้คือบทพิสูจน์แรกของทฤษฎีบทสำคัญที่ต้องอาศัยคอมพิวเตอร์โดยพื้นฐาน และมันทำให้นักคณิตศาสตร์รู้สึกไม่มั่นคง เพราะมันทำลายข้อสมมติที่เก่าแก่มากจนไม่เคยต้องพูดออกมาเลย: ว่าบทพิสูจน์คือสิ่งที่มนุษย์ผู้ผ่านการฝึกฝนสามารถ ในหลักการ ตามอ่านได้ตั้งแต่บรรทัดแรกจนบรรทัดสุดท้าย และรับรองได้ด้วยความเข้าใจของตัวเอง ทฤษฎีบทสี่สีนั้นแทบจะแน่นอนว่าเป็นจริง แต่ข้อโต้แย้งของแอปเพล-ฮาเคนนับเป็นบทพิสูจน์ในความหมายเก่าหรือไม่ — สิ่งที่ surveyable (ตรวจสอบได้ทั้งหมด) สิ่งที่จิตหนึ่งถือครองไว้ได้ — เป็นคำถามที่แวดวงคณิตศาสตร์ไม่เคยปิดได้อย่างสมบูรณ์

That unease is this chapter's opening exhibit, and it names the first principle directly: **surveyability is not guaranteed, and its loss changes what proof can do for us.** Descartes's classical picture of mathematical certainty assumed a chain of steps small enough, and clear enough, for one mind to inspect every link. Appel and Haken's proof is sound by every formal standard computer scientists can specify, yet it asks for a different kind of trust — trust that the compiler compiled correctly, that the hardware did

not glitch across 1,200 hours, that the case-generating program itself contained no error. The theorem may be certain; the *route* to believing it has become an engineering claim as much as a logical one.

> ความไม่มั่นคงนั้นคือหลักฐานชิ้นเปิดของบทนี้ และมันตั้งชื่อหลักการแรกโดยตรง: surveyable ไม่ได้ถูกรับประกัน และการสูญเสียมันเปลี่ยนสิ่งที่บทพิสูจน์ทำเพื่อเราได้ ภาพคลาสสิกของเดส์การ์ตส์เรื่องความแน่นอนทางคณิตศาสตร์สมมติว่ามีลูกโซ่ของขั้นตอนที่เล็กพอและชัดเจนพอ ให้จิตหนึ่งตรวจดูทุกข้อต่อได้ บทพิสูจน์ของแอปเพลกับฮาเคนสมเหตุสมผลตามมาตรฐานรูปนัยทุกข้อที่นักวิทยาการคอมพิวเตอร์ระบุได้ แต่มันเรียกร้องความไว้วางใจอีกแบบหนึ่ง — ความไว้วางใจว่าคอมไพเลอร์คอมไพล์ถูกต้อง ว่าฮาร์ดแวร์ไม่มีอาการสะดุดตลอด 1,200 ชั่วโมงนั้น ว่าโปรแกรมสร้างกรณีเองไม่มีข้อผิดพลาด ทฤษฎีบทนั้นอาจแน่นอน แต่เส้นทางไปสู่การเชื่อมันกลายเป็นข้อกล่าวอ้างเชิงวิศวกรรมพอๆ กับเชิงตรรกะ

The second principle addresses a method that never pretended to surveyability in the first place: simulation. Paul Humphreys and, at greater length, Eric Winsberg (*Science in the Age of Computer Simulation*, University of Chicago Press, 2010) have argued that computer simulation supplies neither classical empirical evidence (you cannot rerun Earth's actual climate as a controlled experiment) nor classical theoretical deduction (a climate model is stitched from theory, numerical approximations, and hand-tuned parameters no single equation justifies). Winsberg calls this a distinct epistemic category: trust in a simulation rests on trust in the model's construction — the physics it encodes, the tricks it uses to make intractable equations tractable, the track record of similar models on cases you *can* check — rather than on trust that the modeled object and the real object are, in the experimenter's sense, "the same kind of system." A hurricane-track simulation is neither an observation of a hurricane nor a proof about one; it is a third thing, evidence built rather than found.

> หลักการที่สองเจาะจงไปที่วิธีการหนึ่งที่ไม่เคยอ้างว่าตรวจสอบได้ทั้งหมดตั้งแต่แรกอยู่แล้ว: การจำลอง (simulation) พอล ฮัมฟรีส์ และเอริก วินส์เบิร์กในรายละเอียดที่มากกว่า (Science in the Age of Computer Simulation, University of Chicago Press, 2010) โต้แย้งว่าการจำลองด้วยคอมพิวเตอร์ไม่ได้ให้ทั้งหลักฐานเชิงประจักษ์

> คลาสสิก (คุณไม่อาจรันสภาพภูมิอากาศจริงของโลกซ้ำในฐานะการทดลองควบคุมได้) หรือการนิรนัยเชิงทฤษฎีคลาสสิก (แบบจำลองภูมิอากาศถูกปะติดปะต่อจากทฤษฎี การประมาณเชิงตัวเลข และพารามิเตอร์ที่ปรับด้วยมือซึ่งไม่มีสมการเดี่ยวใดรับรองได้) วินส์เบิร์กเรียกสิ่งนี้ว่าหมวดหมู่เชิงญาณวิทยาที่แยกต่างหาก: ความไว้วางใจในการจำลองหนึ่งตั้งอยู่บนความไว้วางใจในการสร้างแบบจำลอง — ฟิสิกส์ที่มันเข้ารหัสไว้ กลเม็ดที่มันใช้ทำให้สมการที่แก้ไม่ได้กลายเป็นแก้ได้ ประวัติผลงานของแบบจำลองคล้ายกันในกรณีที่คุณตรวจสอบได้ — มากกว่าจะตั้งอยู่บนความไว้วางใจว่าวัตถุที่ถูกจำลองกับวัตถุจริงเป็น "ระบบชนิดเดียวกัน" ในความหมายของผู้ทดลอง การจำลองเส้นทางเฮอริเคนไม่ใช่ทั้งการสังเกตเฮอริเคนหนึ่งและไม่ใช่บทพิสูจน์เกี่ยวกับมัน มันคือสิ่งที่สาม หลักฐานที่ถูกสร้างขึ้น ไม่ใช่ถูกพบเจอ

The third principle is opacity, and it cuts a line straight through modern computing. Humphreys coined "epistemic opacity" for processes in which not all the elements relevant to a result's justification are, even in principle, available to the epistemic agent making use of it (*Extending Ourselves: Computational Science, Empiricism, and Scientific Method*, Oxford University Press, 2004). A traditionally written program is transparent in a specific sense: however long, its logic is in principle inspectable, line by line, by a competent reader. A trained neural network is opaque in a different sense: its "reasons" are distributed across millions or billions of weighted connections that were never designed, only grown by gradient descent, and no human — not even its authors — can trace why one output rather than another emerged. This bears directly on machine testimony: when a diagnostic model flags a tumor, is that a *report* from a reasoning entity, to be weighed the way we weigh a colleague's judgment, or a *reading*, to be weighed the way we weigh a thermometer's? The two carry different epistemic weight, and opaque systems make it uncomfortably hard to tell which one you are getting.

> หลักการที่สามคือความทึบ (opacity) และมันตัดผ่านวงการคำนวณสมัยใหม่เป็นเส้นตรง ฮัมฟรีส์บัญญัติคำว่าความทึบเชิงญาณวิทยา (epistemic opacity) ขึ้นสำหรับกระบวนการที่องค์ประกอบทั้งหมดซึ่งเกี่ยวข้องกับการให้เหตุผลรองรับผลลัพธ์หนึ่ง ไม่ได้เปิดให้ผู้รู้ (epistemic agent) ที่ใช้มันเข้าถึงได้ แม้ในหลักการก็

> ตาม (Extending Ourselves: Computational Science, Empiricism, and Scientific Method, Oxford University Press, 2004) โปรแกรมที่เขียนแบบดั้งเดิมโปร่งใสในความหมายเฉพาะอย่างหนึ่ง: ไม่ว่าจะยาวแค่ไหน ตรรกะของมันตรวจสอบได้ในหลักการ ทีละบรรทัด โดยผู้อ่านที่มีความสามารถ โครงข่ายประสาทเทียมที่ผ่านการฝึกทึบในความหมายที่ต่างออกไป: "เหตุผล" ของมันกระจายอยู่ทั่วการเชื่อมต่อถ่วงน้ำหนักเป็นล้านหรือพันล้านจุด ที่ไม่เคยถูกออกแบบ มีแต่ถูกปลูกขึ้นด้วย gradient descent และไม่มีมนุษย์คนใด — แม้แต่ผู้สร้างมันเอง — ตามรอยได้ว่าทำไมเอาต์พุตหนึ่งจึงเกิดขึ้นแทนอีกอันหนึ่ง เรื่องนี้เกี่ยวข้องโดยตรงกับคำให้การ (testimony) ของเครื่องจักร: เมื่อแบบจำลองวินิจฉัยหนึ่งตีธงว่าพบเนื้องอก นั่นคือรายงานจากสิ่งที่ให้เหตุผลได้ ที่ควรชั่งน้ำหนักแบบเดียวกับที่เราชั่งน้ำหนักคำตัดสินของเพื่อนร่วมงาน หรือคือการอ่านค่า ที่ควรชั่งน้ำหนักแบบเดียวกับที่เราชั่งน้ำหนักค่าจากเทอร์โมมิเตอร์ ทั้งสองแบบแบกน้ำหนักเชิงญาณวิทยา (epistemic weight) ที่ต่างกัน และระบบทึบทำให้ยากอย่างอึดอัดใจที่จะบอกได้ว่าคุณกำลังได้รับแบบไหนกันแน่

The fourth principle replays an old civil war under new names: program verification versus testing, rationalism versus empiricism. In May 1979, Richard De Millo, Richard Lipton, and Alan Perlis published "Social Processes and Proofs of Theorems and Programs" (*Communications of the* ACM 22, no. 5: 271–280), arguing that mathematical certainty rests on a social process — peers checking, re-deriving, and eventually trusting a proof — that program verification could never replicate at scale, because no comparable community will re-verify a million-line formal proof of a shipped product. James Fetzer's 1988 "Program Verification: The Very Idea" (*Communications of the* ACM 31, no. 9: 1048–1063) sharpened the objection into a philosophical claim: algorithms, as abstract logical structures, admit deductive proof; *programs*, as physical processes running on real, occasionally faulty hardware, do not, because no deduction can certify that silicon will behave as its specification assumes. The verificationist's rationalist ideal — prove it once, know it forever — meets the tester's empiricist reply, distilled by Edsger Dijkstra a decade earlier: "Program testing can be used to show the presence of bugs, but never to show their absence" ("Notes on

Structured Programming," EWD249, 1970). Neither side has won; software ships today on some blend of both, verified where it is cheap enough, tested everywhere else.

> หลักการที่สี่เล่นซ้ำสงครามกลางเมืองเก่าภายใต้ชื่อใหม่: การพิสูจน์ความถูกต้องของโปรแกรม (program verification) กับการทดสอบ เหตุผลนิยม (rationalism) กับประสบการณ์นิยม (empiricism) ในเดือนพฤษภาคม 1979 ริชาร์ด เดอมิลโล ริชาร์ด ลิปตัน และอลัน เพอร์ลิสตีพิมพ์ "Social Processes and Proofs of Theorems and Programs" (Communications of the ACM 22, no. 5: 271–280) โต้แย้งว่าความแน่นอนทางคณิตศาสตร์ตั้งอยู่บนกระบวนการทางสังคม — เพื่อนร่วมวงการตรวจสอบ อนุมานซ้ำ และในที่สุดก็เชื่อบทพิสูจน์หนึ่ง — ซึ่งการพิสูจน์ความถูกต้องของโปรแกรมไม่มีทางทำซ้ำได้ในระดับใหญ่ เพราะไม่มีชุมชนที่เทียบเคียงได้จะมาตรวจสอบซ้ำบทพิสูจน์รูปนัยยาวล้านบรรทัดของผลิตภัณฑ์ที่ส่งมอบแล้ว "Program Verification: The Very Idea" ของเจมส์ เฟตเซอร์ปี 1988 (Communications of the ACM 31, no. 9: 1048–1063) ลับข้อคัดค้านนั้นให้คมขึ้นเป็นข้อกล่าวอ้างเชิงปรัชญา: อัลกอริทึม ในฐานะโครงสร้างเชิงตรรกะนามธรรม ยอมรับบทพิสูจน์เชิงนิรนัยได้ แต่โปรแกรม ในฐานะกระบวนการทางกายภาพที่รันบนฮาร์ดแวร์จริงซึ่งบางครั้งมีข้อบกพร่อง ไม่ยอมรับ เพราะไม่มีการนิรนัยใดรับรองได้ว่าซิลิคอนจะทำตัวตามที่ข้อกำหนดสมมติไว้ อุดมคติแบบเหตุผลนิยมของฝ่ายพิสูจน์ความถูกต้อง — พิสูจน์ครั้งเดียว รู้ตลอดไป — มาเจอกับคำตอบโต้แบบประสบการณ์นิยมของฝ่ายทดสอบ ที่เอ็ดสเกอร์ ไดก์สตราสรุปไว้หนึ่งทศวรรษก่อนหน้านั้นว่า "การทดสอบโปรแกรมใช้แสดงการมีอยู่ของบั๊กได้ แต่ไม่เคยใช้แสดงการไม่มีอยู่ของมันได้เลย" ("Program testing can be used to show the presence of bugs, but never to show their absence") ("Notes on Structured Programming," EWD249, 1970) ไม่มีฝ่ายใดชนะ ซอฟต์แวร์ทุกวันนี้ส่งมอบด้วยส่วนผสมของทั้งสองแบบ พิสูจน์ความถูกต้องตรงที่ทำได้ถูกพอ ทดสอบในที่อื่นๆ ทั้งหมด

The fifth principle is the oldest trap in epistemology, wearing a sensor's housing. Edmund Gettier's three-page 1963 paper, "Is Justified True Belief Knowledge?" (*Analysis* 23, no. 6: 121–123), showed that a belief can be justified and true and still fail to be knowledge, when the justification and the truth connect only by luck — the classic image is a stopped clock that hap-

pens to show the correct time. Automated pipelines manufacture this structure routinely: a facial-recognition system with a genuine bug — miscalibrated, say, on faces at a certain angle — can still output the correct match on a given frame, by accident, while carrying all the institutional justification ("this system is 99% accurate") that would normally certify a belief as knowledge. The output is true, and reasonably trusted, and arrived at by a process that, on that occasion, was not doing what its justification assumed it was doing. Gettier's stopped clock now has cameras.

> หลักการที่ห้าคือกับดักที่เก่าแก่ที่สุดในญาณวิทยา สวมเปลือกของเซนเซอร์หนึ่ง บทความสามหน้าของเอ็ดมันด์ เก็ตติเยร์ปี 1963 "Is Justified True Belief Knowledge?" (Analysis 23, no. 6: 121–123) แสดงให้เห็นว่าความเชื่อหนึ่งอาจมีเหตุผลรองรับและเป็นจริง แต่ก็ยังไม่ใช่ความรู้ได้ เมื่อเหตุผลรองรับกับความจริงเชื่อมกันแค่ด้วยความบังเอิญ — ภาพคลาสสิกคือนาฬิกาที่หยุดเดินแล้วบังเอิญบอกเวลาถูกต้อง สายพานอัตโนมัติผลิตโครงสร้างแบบนี้ขึ้นเป็นประจำ: ระบบจดจำใบหน้าที่มีบั๊กจริง — สมมติว่าปรับเทียบผิดสำหรับใบหน้ามุมหนึ่ง — ยังคงให้ผลจับคู่ที่ถูกต้องบนเฟรมหนึ่งได้ โดยบังเอิญ ในขณะที่แบกเหตุผลรองรับเชิงสถาบันทั้งหมด ("ระบบนี้แม่นยำ 99%") ที่ปกติแล้วจะรับรองความเชื่อหนึ่งให้เป็นความรู้ได้ ผลลัพธ์นั้นเป็นจริง และถูกไว้วางใจอย่างสมเหตุสมผล และมาถึงด้วยกระบวนการที่ ในโอกาสนั้น ไม่ได้ทำตามที่เหตุผลรองรับของมันสมมติไว้ว่ามันกำลังทำ นาฬิกาที่หยุดเดินของเก็ตติเยร์ตอนนี้มีกล้องแล้ว

The people, briefly. Appel and Haken took considerable heat for a proof many still call inelegant rather than false. Humphreys and Winsberg built an entire subfield — the philosophy of computer simulation — out of taking seriously a method working scientists had already trusted for decades without asking what kind of evidence it was. Fetzer's 1988 paper triggered

pages of technical correspondence within a year and a debate still cited today — one of the longest-running controversies *Communications of the* ACM ever printed.

> **ผู้คน โดยย่อ** แอปเพลกับฮาเคนโดนวิจารณ์หนักพอสมควรสำหรับบทพิสูจน์ที่หลายคนยังคงเรียกว่าไม่สง่างามมากกว่าจะเรียกว่าผิด ฮัมฟรีส์กับวินส์เบิร์กสร้างสาขาย่อยทั้งสาขาขึ้นมา — ปรัชญาแห่งการจำลองด้วยคอมพิวเตอร์ — จากการเอาจริงเอาจังกับวิธีการหนึ่งที่นักวิทยาศาสตร์ที่ทำงานจริงไว้วางใจมาหลายสิบปีแล้ว โดยไม่เคยถามว่ามันเป็นหลักฐานชนิดไหนกันแน่ บทความปี 1988 ของเฟตเซอร์จุดชนวนจดหมายโต้ตอบเชิงเทคนิคหลายหน้าภายในหนึ่งปี และการถกเถียงที่ยังถูกอ้างอิงถึงทุกวันนี้ — หนึ่งในข้อขัดแย้งที่ยืนยาวที่สุดที่ Communications of the ACM เคยตีพิมพ์

Connections · เชื่อมโยง

Chapter 01 supplies the proof-theoretic background — soundness, completeness, and Gödel's limits — that surveyability quietly assumes. Chapter 09 asks what undecidability licenses philosophically, a companion question to what unsurveyable proof licenses here. Chapter 13 returns to opacity from the learning side: interpretability research as a laboratory for finding out what, if anything, is inside the black box.

> บทที่ 01 ให้พื้นหลังเชิงทฤษฎีบทพิสูจน์ — ความสมเหตุสมผลเชิงตรรกะ ความสมบูรณ์ และขีดจำกัดของเกอเดิล — ที่ surveyability สมมติไว้อย่างเงียบๆ บทที่ 09 ถามว่าการตัดสินไม่ได้ (undecidability) อนุญาตให้อ้างอะไรได้บ้างในเชิงปรัชญา คำถามคู่กับสิ่งที่บทพิสูจน์ที่ตรวจสอบไม่ได้ทั้งหมดอนุญาตให้อ้างที่นี่ บทที่ 13 กลับไปที่ความทึบจากฝั่งการเรียนรู้: งานวิจัย interpretability ในฐานะห้องทดลองสำหรับค้นหาว่ามีอะไรอยู่ข้างในกล่องดำบ้าง ถ้ามี

Open questions · คำถามที่ยังเปิดอยู่

If a proof can be certain but not surveyable, has "proof" quietly changed meaning since Euclid, or did it always mean "warranted to the standards of the day" and Euclid's standards were simply stricter than ours by accident

of history? And is machine opacity a temporary engineering limitation interpretability research will eventually dissolve, or a permanent structural feature of any system whose "knowledge" is distributed statistically rather than written down as rules?

> ถ้าบทพิสูจน์หนึ่งแน่นอนได้แต่ตรวจสอบได้ทั้งหมดไม่ได้ คำว่า "บทพิสูจน์" เปลี่ยนความหมายไปอย่างเงียบๆ ตั้งแต่ยุคยูคลิดหรือไม่ หรือมันหมายถึง "ได้รับการรับรองตามมาตรฐานของยุคนั้น" มาตลอด และมาตรฐานของยูคลิดก็แค่เข้มงวดกว่าของเราด้วยความบังเอิญทางประวัติศาสตร์ และความทึบของเครื่องจักรเป็นข้อจำกัดเชิงวิศวกรรมชั่วคราวที่งานวิจัย interpretability จะสลายไปได้ในที่สุด หรือเป็นคุณลักษณะเชิงโครงสร้างถาวรของระบบใดๆ ที่ "ความรู้" ของมันกระจายอยู่เชิงสถิติแทนที่จะถูกเขียนไว้เป็นกฎ

Go deeper · อ่านต่อ

- Eric Winsberg, *Science in the Age of Computer Simulation*, University of Chicago Press, 2010 — the founding case for simulation as its own epistemic category.

  > Eric Winsberg, Science in the Age of Computer Simulation, University of Chicago Press, 2010 — ข้อเสนอก่อตั้งสำหรับการจำลองในฐานะหมวดหมู่เชิงญาณวิทยาของตัวมันเอง

- Richard A. De Millo, Richard J. Lipton, and Alan J. Perlis, "Social Processes and Proofs of Theorems and Programs," *Communications of the ACM* 22, no. 5 (1979): 271–280; James H. Fetzer, "Program Verification: The Very Idea," *Communications of the ACM* 31, no. 9 (1988): 1048–1063.

  > Richard A. De Millo, Richard J. Lipton, and Alan J. Perlis, "Social Processes and Proofs of Theorems and Programs," Communications of the ACM 22, no. 5 (1979): 271–280; James H. Fetzer, "Program Verification: The Very Idea," Communications of the ACM 31, no. 9 (1988): 1048–1063

- *Computer Simulations in Science*, Stanford Encyclopedia of Philosophy.

  > Computer Simulations in Science, สารานุกรมปรัชญาสแตนฟอร์ด (SEP)

CHAPTER 04

Philosophy of Mind

ปรัชญาแห่งจิต

Part I's fourth hearing: metaphysics asked what a program is; this chapter asks whether that same kind of thing could be what you are.

การไต่สวนครั้งที่สี่ของภาค I: อภิปรัชญาถามว่าโปรแกรมคืออะไร บทนี้ถามต่อว่าสิ่งชนิดเดียวกันนั้นอาจเป็นสิ่งที่คุณเป็นได้หรือไม่

Is the mind what the brain computes?

จิตคือสิ่งที่สมองคำนวณหรือไม่

In 1980, John Searle asked readers of *Behavioral and Brain Sciences* to imagine him locked in a room with a rulebook written in English and a slot through which cards covered in Chinese characters arrive. Searle does not read Chinese. But the rulebook is good enough — for any incoming string of characters, it tells him, by shape alone, which characters to pass back out. To a Chinese speaker outside, the answers look fluent, even witty; the room, treated as a black box, could pass a Turing test conducted in Chinese. Searle's question was simple and became one of the most argued-over pages in twentieth-century philosophy: does anyone, or anything, in that room *understand* Chinese? His answer was no. A person manipulating symbols by their shape, however successfully, is not thereby extracting their meaning — and if that is true of a man with a rulebook, it is true of a computer running the equivalent program, however large. Searle called the

paper "Minds, Brains, and Programs" (*Behavioral and Brain Sciences* 3, 1980: 417–457); the journal ran it in its open-peer-commentary format, its objectors printed alongside, and the arguing has not really stopped since.

> ในปี 1980 จอห์น เซิร์ลขอให้ผู้อ่าน Behavioral and Brain Sciences จินตนาการว่าตัวเขาถูกขังอยู่ในห้องหนึ่งพร้อมกับสมุดกฎที่เขียนเป็นภาษาอังกฤษ และช่องหนึ่งที่บัตรเต็มไปด้วยตัวอักษรจีนถูกส่งเข้ามา เซิร์ลอ่านภาษาจีนไม่ออก แต่สมุดกฎนั้นดีพอ — สำหรับสายอักขระที่ส่งเข้ามาใดๆ มันบอกเขาได้ ด้วยรูปร่างล้วนๆ ว่าจะส่งตัวอักษรไหนกลับออกไป สำหรับผู้พูดภาษาจีนที่อยู่ข้างนอก คำตอบดูคล่องแคล่ว แม้กระทั่งมีไหวพริบ ห้องนั้น เมื่อถูกปฏิบัติเป็นกล่องดำ ก็ผ่านการทดสอบทัวริง (Turing test) ที่ทำเป็นภาษาจีนได้ คำถามของเซิร์ลเรียบง่ายและกลายเป็นหนึ่งในหน้าที่ถูกถกเถียงมากที่สุดในปรัชญาศตวรรษที่ยี่สิบ: มีใครหรือสิ่งใดในห้องนั้นเข้าใจภาษาจีนหรือไม่ คำตอบของเขาคือไม่มี คนหนึ่งที่จัดการสัญลักษณ์ตามรูปร่างของมัน ไม่ว่าจะสำเร็จแค่ไหน ก็ไม่ได้สกัดความหมายของมันออกมาด้วยเหตุนั้น — และถ้าเรื่องนี้เป็นจริงสำหรับชายคนหนึ่งกับสมุดกฎ มันก็เป็นจริงสำหรับคอมพิวเตอร์ที่รันโปรแกรมเทียบเท่ากัน ไม่ว่าจะใหญ่แค่ไหนก็ตาม เซิร์ลเรียกบทความนี้ว่า "Minds, Brains, and Programs" (Behavioral and Brain Sciences 3, 1980: 417–457) วารสารตีพิมพ์มันในรูปแบบวิจารณ์เปิดเผยโดยเพื่อนร่วมวงการ พิมพ์ฝ่ายคัดค้านไว้ควบคู่กัน และการถกเถียงก็ไม่เคยหยุดจริงๆ นับแต่นั้นมา

The Chinese Room was aimed at a specific, then-dominant target, which is this chapter's first principle: the computational theory of mind. Hilary Putnam's papers from 1960 onward, culminating in "The Nature of Mental States" (1967), proposed that a mental state is not defined by what it is made of — neurons, silicon, anything — but by its functional role: what causes it, what it causes, how it relates to other mental states. His guiding picture was the Turing machine, whose abstract "machine table" of states is exactly the same whether it runs on relays, vacuum tubes, or a napkin and a pencil, so long as the transitions are honored. Jerry Fodor pushed the theory from an analogy into an architecture in *The Language of Thought* (1975): thinking, he argued, is computation over an internal symbolic code — "Mentalese" — with a combinatorial syntax and a compositional semantics, exactly the kind of structure a digital computer manipulates. On this view, a mind literally is

a program running on the wet hardware of a brain, and Searle's room is the theory's most inconvenient test case: a system that, on Putnam and Fodor's own terms, is running the right program, and yet — Searle insists — plainly understands nothing.

> ห้องจีน (Chinese Room) เล็งเป้าไปที่เป้าหมายเฉพาะเจาะจงที่ครองความโดดเด่นในตอนนั้น ซึ่งเป็นหลักการแรกของบทนี้: ทฤษฎีจิตเชิงคำนวณ (computational theory of mind) บทความของฮิลารี พัทนัมตั้งแต่ปี 1960 เป็นต้นมา จบลงที่ "The Nature of Mental States" (1967) เสนอว่าสถานะจิตหนึ่งไม่ได้ถูกนิยามด้วยสิ่งที่มันทำขึ้นมา — เซลล์ประสาท ซิลิคอน อะไรก็ตาม — แต่ด้วยบทบาทเชิงหน้าที่ (functional role) ของมัน: อะไรเป็นเหตุให้เกิดมัน มันเป็นเหตุให้เกิดอะไร มันสัมพันธ์กับสถานะจิตอื่นอย่างไร ภาพนำทางของเขาคือเครื่องทัวริง ที่ "ตารางเครื่อง" นามธรรมของสถานะต่างๆ เหมือนกันเป๊ะไม่ว่าจะรันบนรีเลย์ หลอดสุญญากาศ หรือกระดาษเช็ดปากกับดินสอ ตราบใดที่การเปลี่ยนสถานะยังคงถูกต้อง เจอร์รี โฟดอร์ผลักทฤษฎีนี้จากการเปรียบเทียบให้กลายเป็นสถาปัตยกรรมใน The Language of Thought (1975): เขาโต้แย้งว่าการคิดคือการคำนวณเหนือรหัสสัญลักษณ์ภายใน — "Mentalese" — ที่มีวากยสัมพันธ์เชิงการจัดหมู่และความหมายเชิงประกอบ (compositionality) เป๊ะๆ เหมือนโครงสร้างที่คอมพิวเตอร์ดิจิทัลจัดการอยู่ ในมุมมองนี้ จิตหนึ่งเป็นโปรแกรมที่รันอยู่บนฮาร์ดแวร์เปียกของสมองโดยแท้จริง และห้องของเซิร์ลคือกรณีทดสอบที่ลำบากใจที่สุดของทฤษฎีนี้: ระบบหนึ่งที่ ตามเงื่อนไขของพัทนัมกับโฟดอร์เอง กำลังรันโปรแกรมที่ถูกต้อง แต่กลับ — เซิร์ลยืนยัน — ไม่เข้าใจอะไรเลยอย่างชัดเจน

The second principle sits one level below that metaphysical claim, and it is easy to confuse with it: Alan Turing's 1950 proposal, in "Computing Machinery and Intelligence" (*Mind* 59, no. 236: 433–460), that the question "can machines think" is too laden with undefined terms to argue productively, and should be replaced by an operational one — can a machine's conversational behavior be mistaken for a human's, in a structured game of questions and answers? Turing was not claiming that passing the game proves the presence of a mind; he was proposing to stop asking the metaphysical question and start measuring the behavioral one. The distance between Turing's move and Putnam and Fodor's is exactly the distance the

Chinese Room exploits: Turing offers a test for *behavior*; the computational theory of mind claims that the *right* behavior-producing process, whatever it turns out to be, simply is thinking; Searle's room passes the first while, he argues, refuting the second.

> หลักการที่สองอยู่ต่ำกว่าข้อกล่าวอ้างเชิงอภิปรัชญานั้นหนึ่งระดับ และสับสนกับมันได้ง่าย: ข้อเสนอปี 1950 ของอลัน ทัวริง ใน "Computing Machinery and Intelligence" (Mind 59, no. 236: 433–460) ที่ว่าคำถาม "เครื่องจักรคิดได้ไหม" แบกคำที่ไม่ได้นิยามไว้มากเกินกว่าจะถกเถียงได้อย่างมีผล และควรถูกแทนที่ด้วยคำถามเชิงปฏิบัติการ — พฤติกรรมการสนทนาของเครื่องจักรหนึ่งถูกเข้าใจผิดว่าเป็นของมนุษย์ได้หรือไม่ ในเกมคำถามคำตอบที่มีโครงสร้าง ทัวริงไม่ได้อ้างว่าการผ่านเกมนี้พิสูจน์การมีอยู่ของจิตหนึ่ง เขาเสนอให้หยุดถามคำถามเชิงอภิปรัชญาและเริ่มวัดคำถามเชิงพฤติกรรมแทน ระยะห่างระหว่างท่าทีของ
> ทัวริงกับของพัทนัมและโฟดอร์ คือระยะห่างเดียวกับที่ห้องจีนใช้ประโยชน์: ทัวริงเสนอการทดสอบสำหรับพฤติกรรม ทฤษฎีจิตเชิงคำนวณอ้างว่ากระบวนการที่ผลิตพฤติกรรมที่ถูกต้อง ไม่ว่าในที่สุดมันจะกลายเป็นอะไร ก็คือการคิดโดยตรง ห้องของเซิร์ลผ่านข้อแรก ในขณะที่ เขาโต้แย้ง มันหักล้างข้อที่สอง

The third principle is the argument's afterlife: the replies, and what they reveal. The "systems reply" holds that Searle is right that the man alone understands nothing, but wrong to conclude that no one does — understanding, on this view, belongs to the whole system: man, rulebook, and paper together. Searle's counter was to have the man memorize the entire rulebook and work the transformations in his head, leaving no separate "paper" for understanding to hide in; by his lights, still no understanding, only a faster clerk. The "robot reply" tries a different repair: wire the room to cameras and motors, so the symbols are caused by, and cause changes in, a real world instead of arriving from and returning to another room. Searle's response — the symbols are still just more symbols, now merely triggered by transducers rather than slips of paper — names, without quite solving, what

later literature calls the symbol grounding problem: rules connecting symbols to other symbols, however elaborate, may never on their own reach the world the symbols are supposed to be about.

> หลักการที่สามคือชีวิตหลังความตายของข้อโต้แย้งนี้: คำตอบโต้ต่างๆ และสิ่งที่มันเผยออกมา "คำตอบโต้เชิงระบบ" (systems reply) ยืนยันว่าเซิร์ลพูดถูกที่ว่าชายคนนั้นคนเดียวไม่เข้าใจอะไรเลย แต่ผิดที่สรุปว่าไม่มีใครเข้าใจเลย — ความเข้าใจ ในมุมมองนี้ เป็นของระบบทั้งหมด: ชายคนนั้น สมุดกฎ และกระดาษ รวมกัน คำตอบโต้ของเซิร์ลคือให้ชายคนนั้นท่องจำสมุดกฎทั้งเล่ม แล้วทำการแปลงในหัวของเขาเอง ไม่เหลือ "กระดาษ" แยกต่างหากให้ความเข้าใจไปซ่อนอยู่ ตามทัศนะของเขา ก็ยังคงไม่มีความเข้าใจอยู่ดี มีแค่เสมียนที่เร็วขึ้น "คำตอบโต้เชิงหุ่นยนต์" (robot reply) ลองซ่อมแซมอีกแบบ: ต่อสายห้องเข้ากับกล้องและมอเตอร์ ให้สัญลักษณ์ถูกก่อขึ้นโดยและก่อให้เกิดการเปลี่ยนแปลงใน โลกจริง แทนที่จะมาจากและกลับไปยังอีกห้องหนึ่ง คำตอบของเซิร์ล — สัญลักษณ์ก็ยังเป็นแค่สัญลักษณ์เพิ่มเติมอยู่ดี ตอนนี้แค่ถูกกระตุ้นด้วยตัวแปลงสัญญาณแทนเศษกระดาษ — ตั้งชื่อ โดยไม่ได้แก้ปัญหาไปเสียทีเดียว สิ่งที่วรรณกรรมภายหลังเรียกว่าปัญหาการยึดโยงสัญลักษณ์ (symbol grounding problem): กฎที่เชื่อมสัญลักษณ์เข้ากับสัญลักษณ์อื่น ไม่ว่าจะซับซ้อนแค่ไหน อาจไม่มีทางไปถึงโลกที่สัญลักษณ์เหล่านั้นควรจะพูดถึงได้ด้วยตัวมันเอง

The fourth principle is an older civil war that current AI reopened rather than settled. Fodor, together with Zenon Pylyshyn, turned his defense of the language of thought into a direct attack on the rival architecture then rising in cognitive science: connectionism, the parallel distributed processing (PDP) networks of David Rumelhart and James McClelland's 1986 volumes. "Connectionism and Cognitive Architecture: A Critical Analysis" (*Cognition* 28, 1988: 3–71) argued that thought is systematic in a way pure connectionist networks cannot, in principle, explain: anyone who can think *John loves Mary* can think *Mary loves John* — the two thoughts share combinatorial parts that recombine, which requires, they claimed, an underlying symbolic syntax, not a pattern-matcher trained on enough examples of both shapes. For thirty years the argument looked mostly settled toward hybrid views. Large language models reopened it at a scale nobody in 1988 anticipated: systems built from nothing but weighted connections now pro-

duce output systematic enough to fool most casual tests, with no hand-built symbolic layer in sight — and the field is still arguing — empirically now, not only philosophically — whether that refutes Fodor and Pylyshyn or whether large networks quietly learn something symbol-like after all, a question chapter 13 picks up.

> หลักการที่สี่คือสงครามกลางเมืองเก่าแก่กว่านั้น ที่ AI ยุคปัจจุบันเปิดขึ้นใหม่มากกว่า จะยุติมันลง โฟดอร์ ร่วมกับเซนอน ไพลีชิน เปลี่ยนการป้องกันภาษาแห่งความคิดของเขาให้กลายเป็นการโจมตีตรงต่อสถาปัตยกรรมคู่แข่งที่กำลังรุ่งเรืองในวิทยาศาสตร์การรู้คิดตอนนั้น: คอนเนคชันนิสม์ (connectionism) เครือข่ายประมวลผลกระจายแบบขนาน (PDP) ในหนังสือสองเล่มปี 1986 ของเดวิด รูเมลฮาร์ตกับเจมส์ แม็คเคลแลนด์ "Connectionism and Cognitive Architecture: A Critical Analysis" (Cognition 28, 1988: 3–71) โต้แย้งว่าความคิดมี systematicity (ความเป็นระบบของความคิด) ในแบบที่เครือข่ายคอนเนคชันนิสม์บริสุทธิ์อธิบายไม่ได้ในหลักการ: ใครก็ตามที่คิดได้ว่า John loves Mary ก็คิดได้ว่า Mary loves John — ความคิดทั้งสองแบ่งปันส่วนที่จัดหมู่ใหม่ได้ ซึ่งพวกเขาอ้างว่าต้องอาศัยวากยสัมพันธ์เชิงสัญลักษณ์อยู่เบื้องหลัง ไม่ใช่ตัวจับคู่แบบรูปที่ถูกเทรนมาด้วยตัวอย่างของทั้งสองรูปแบบมากพอ สามสิบปีที่ผ่านมาข้อโต้แย้งนี้ดูเหมือนจะยุติลงไปทางมุมมองแบบผสมเป็นส่วนใหญ่ โมเดลภาษาขนาดใหญ่ (large language models) เปิดมันขึ้นใหม่ในสเกลที่ไม่มีใครในปี 1988 คาดการณ์ไว้: ระบบที่สร้างขึ้นจากการเชื่อมต่อถ่วงน้ำหนักล้วนๆ ตอนนี้ผลิตเอาต์พุตที่มี systematicity มากพอจะหลอกการทดสอบทั่วไปส่วนใหญ่ได้ โดยไม่มีชั้นสัญลักษณ์ที่สร้างด้วยมือให้เห็นเลย — และวงการยังคงถกเถียงอยู่ — ตอนนี้เชิงประจักษ์ ไม่ใช่แค่เชิงปรัชญาอีกต่อไป — ว่าสิ่งนี้หักล้างโฟดอร์กับไพลีชินหรือไม่ หรือเครือข่ายขนาดใหญ่แอบเรียนรู้บางสิ่งที่คล้ายสัญลักษณ์ไปด้วยในที่สุด คำถามที่บทที่ 13 หยิบขึ้นมาต่อ

The fifth principle is the one no reply above touches: David Chalmers's distinction, first presented at the 1994 Tucson conference on consciousness and published as "Facing Up to the Problem of Consciousness" (*Journal of Consciousness Studies* 2, no. 3, 1995: 200–219), between the *easy* problems of mind — explaining discrimination, integration, reportability, attention, all of them tractable, in principle, by computational or neural mechanism — and

the *hard* problem: why any of that processing is accompanied by subjective experience at all, why there is something it is like to be the system doing it. A computational theory of mind, however complete, explains structure and function; Chalmers argues it can stay silent on why function comes with feeling, illustrated by his "zombie" duplicate — functionally identical to a conscious being, with no inner experience at all. Whatever one makes of zombies, the hard problem marks where this chapter's apparatus — programs, functions, computation — may run out of the right kind of thing to explain. What current AI revives is all four earlier principles at once, unresolved, at a scale nobody argued at before: the field keeps measuring behavior, because behavior is what it knows how to measure, while the Chinese Room's question has not gone anywhere.

> หลักการที่ห้าคือหลักการเดียวที่ไม่มีคำตอบโต้ข้างต้นแตะถึง: ความแตกต่างของ เดวิด ชาลเมอร์ส ที่นำเสนอครั้งแรกในการประชุมทูซอนปี 1994 ว่าด้วยจิตสำนึก และตีพิมพ์เป็น "Facing Up to the Problem of Consciousness" (Journal of Consciousness Studies 2, no. 3, 1995: 200–219) ระหว่างปัญหาง่าย (the easy problems) ของจิต — การอธิบายการแยกแยะ การรวมเข้าด้วยกัน ความสามารถในการรายงานได้ ความใส่ใจ ทั้งหมดนี้จัดการได้ในหลักการด้วยกลไกเชิงคำนวณหรือเชิงประสาท — กับปัญหายาก (the hard problem): ทำไมกระบวนการเหล่านั้นถึงมาพร้อมกับประสบการณ์เชิงอัตวิสัย (subjective experience) เลยด้วยซ้ำ ทำไมถึงมี "ความรู้สึกว่าเป็นอย่างไร" ที่จะเป็นระบบที่กำลังทำสิ่งนั้นอยู่ ทฤษฎีจิตเชิงคำนวณไม่ว่าจะสมบูรณ์แค่ไหน ก็อธิบายโครงสร้างกับหน้าที่ได้ ชาลเมอร์สโต้แย้งว่ามันนิ่งเงียบได้ต่อคำถามว่าทำไมหน้าที่ถึงมาพร้อมความรู้สึก แสดงให้เห็นด้วยแบบจำลองซ้ำ "ซอมบี้" ของเขา — ซอมบี้เชิงปรัชญา (philosophical zombie) ที่เหมือนกันทุกประการเชิงหน้าที่กับสิ่งมีจิตสำนึก แต่ไม่มีประสบการณ์ภายในเลย ไม่ว่าใครจะคิดอย่างไรกับซอมบี้ ปัญหายากนี้ทำเครื่องหมายจุดที่กลไกของบทนี้ — โปรแกรม ฟังก์ชัน การคำนวณ — อาจหมดสิ่งที่ถูกชนิดพอจะอธิบายได้ ปัญญาประดิษฐ์ยุคปัจจุบันฟื้นคืนทั้งสี่หลักการก่อนหน้าพร้อมกัน โดยยังไม่ได้รับการแก้ไข ในสเกลที่ไม่มีใครเคยถกเถียงมาก่อน: วงการยังคงวัดพฤติกรรมต่อไป เพราะพฤติกรรมคือสิ่งที่มันรู้วิธีวัด ในขณะที่คำถามของห้องจีนยังไม่ไปไหนเลย

The people, briefly. Putnam abandoned computational functionalism before many of his own students did, uneasy with a theory that could not survive his own triviality argument from chapter 02. Fodor defended the language of thought against successive rivals for forty years and outlived most of them. Searle has never claimed computers cannot in principle think — his target is the *sufficiency* of syntax, not the possibility of machine minds by other means. Chalmers named the hard problem in his twenties and has spent every year since being asked to solve it.

> **ผู้คน โดยย่อ** พัทนัมทิ้งหน้าที่นิยมเชิงคำนวณก่อนที่ลูกศิษย์ของเขาเองหลายคนจะทำเสียอีก รู้สึกไม่สบายใจกับทฤษฎีหนึ่งที่รอดจากข้อโต้แย้งเรื่องความจิ๊บจ๊อยของตัวเขาเองในบทที่ 02 ไม่ได้ โฟดอร์ปกป้องภาษาแห่งความคิดจากคู่แข่งที่ผลัดกันมาสี่สิบปี และอยู่รอดยาวนานกว่าคู่แข่งส่วนใหญ่ เซิร์ลไม่เคยอ้างว่าคอมพิวเตอร์คิดไม่ได้ในหลักการ — เป้าหมายของเขาคือความเพียงพอของวากยสัมพันธ์ ไม่ใช่ความเป็นไปได้ของจิตเครื่องจักรด้วยวิธีอื่น ชาลเมอร์สตั้งชื่อปัญหายากตอนอายุยี่สิบกว่า และใช้ทุกปีนับแต่นั้นถูกขอให้แก้มัน

Connections · เชื่อมโยง

Chapter 02's implementation problem is this chapter's silent premise: if implementing a program does not settle what a system is computing, functionalism inherits that gap directly. Chapter 08 returns to embodiment from Dreyfus's angle, against which the robot reply is a partial, contested answer. Chapter 09 owns the Lucas–Penrose argument, which aims Gödel's limits at this chapter's thesis from outside. Chapter 13 owns the full empirical run of connectionism versus symbols, grounding, and what interpretability finds inside the systems this chapter can only interrogate from outside.

> ปัญหาการนำไปใช้จริงของบทที่ 02 คือข้อสมมติเงียบของบทนี้: ถ้าการนำโปรแกรมไปใช้จริงไม่ได้ยุติว่าระบบหนึ่งกำลังคำนวณอะไร หน้าที่นิยมก็รับช่วงช่องว่างนั้นโดยตรง บทที่ 08 กลับไปที่ embodied cognition (การรู้คิดแบบมีร่าง) จากมุมของเดรย์ฟัส ซึ่งคำตอบโต้เชิงหุ่นยนต์เป็นคำตอบบางส่วนที่ยังมีข้อโต้แย้งอยู่ บทที่ 09 เป็นเจ้าของข้อโต้แย้งลูคัส–เพนโรส ที่เล็งขีดจำกัดของเกอเดิลมาที่วิทยานิพนธ์ของ

> บทนี้จากภายนอก บทที่ 13 เป็นเจ้าของการทดลองเชิงประจักษ์เต็มรูปแบบของคอนเนคชันนิสม์เทียบกับสัญลักษณ์ การยึดโยง และสิ่งที่ interpretability พบข้างในระบบที่บทนี้ทำได้แค่ซักถามจากภายนอกเท่านั้น

Open questions · คำถามที่ยังเปิดอยู่

If Turing's operational test and Searle's understanding-requirement are both individually reasonable, can they be held at once — is passing the imitation game evidence of thought, weak evidence, or simply orthogonal to it? And if the hard problem is genuinely intractable to computational explanation, what follows for how we ought to treat systems whose behavior increasingly resembles a mind's, when behavior is the only thing any test — Turing's included — was ever built to measure?

> ถ้าการทดสอบเชิงปฏิบัติการของทัวริงกับข้อเรียกร้องเรื่องความเข้าใจของเซิร์ล ต่างสมเหตุสมผลเมื่อพิจารณาแยกกัน คนเราถือทั้งสองไว้พร้อมกันได้หรือไม่ — การผ่านเกมเลียนแบบเป็นหลักฐานของการคิด หลักฐานที่อ่อน หรือเป็นเรื่องที่ไม่เกี่ยวข้องกันเลยกันแน่ และถ้าปัญหายากอธิบายไม่ได้จริงๆ ด้วยคำอธิบายเชิงคำนวณ อะไรจะตามมาสำหรับวิธีที่เราควรปฏิบัติต่อระบบที่พฤติกรรมของมันคล้ายจิตหนึ่งมากขึ้นเรื่อยๆ ในเมื่อพฤติกรรมคือสิ่งเดียวที่การทดสอบใดๆ — รวมถึงของทัวริงด้วย — เคยถูกสร้างขึ้นมาเพื่อวัด

Go deeper · อ่านต่อ

— Margaret A. Boden, *Mind as Machine*: A *History of Cognitive Science*, 2 vols., Oxford University Press, 2006 — the field's own history, told by one of its central participants.

> Margaret A. Boden, Mind as Machine: A History of Cognitive Science, 2 vols., Oxford University Press, 2006 — ประวัติศาสตร์ของวงการเอง เล่าโดยหนึ่งในผู้มีบทบาทหลัก

— John R. Searle, "Minds, Brains, and Programs," *Behavioral and Brain Sciences* 3 (1980): 417–457; Jerry A. Fodor and Zenon W. Pylyshyn, "Connectionism and Cognitive Architecture: A Critical Analysis," *Cognition* 28 (1988): 3–71.

> John R. Searle, "Minds, Brains, and Programs," Behavioral and Brain Sciences 3 (1980): 417–457; Jerry A. Fodor and Zenon W. Pylyshyn, "Connectionism and Cognitive Architecture: A Critical Analysis," Cognition 28 (1988): 3–71

— *The Computational Theory of Mind*, Stanford Encyclopedia of Philosophy.

> The Computational Theory of Mind, สารานุกรมปรัชญาสแตนฟอร์ด (SEP)

CHAPTER 05

Philosophy of Language

ปรัชญาภาษา

This chapter sits in Part I, the Court of Philosophy — the fifth hearing, where philosophy asks the machine what its code, and our own speech, actually mean.

บทนี้อยู่ในภาค I ศาลปรัชญา — การไต่สวนครั้งที่ 5 ที่ปรัชญาตั้งคำถามกับเครื่องจักรว่าโค้ดของมัน และคำพูดของเราเอง มีความหมายว่าอะไรกันแน่

How does code mean — and how do we?

โค้ดมีความหมายได้อย่างไร — แล้วเรามีความหมายได้อย่างไร

On 17 June 2016, an anonymous attacker called a function named `splitDAO` inside The DAO, a decentralized investment fund that had raised over $150 million in ether from thousands of contributors that spring. The function was supposed to let a member withdraw their share and then update the fund's internal ledger — but it updated the ledger *after* sending the funds, not before, so the attacker called it again from inside the same transaction, before the balance had changed, and kept calling it in a tight recursive loop. By the time the loop stopped, roughly 3.6 million ether — around $50 million by the day's reporting — had drained into an account the attacker controlled (Chainlink, "Reentrancy Attacks and The DAO Hack Explained"). Nothing the attacker did violated a single line of the contract's code. Every call matched every rule exactly.

> วันที่ 17 มิถุนายน 2016 ผู้โจมตีนิรนามคนหนึ่งเรียกใช้ฟังก์ชันชื่อ `splitDAO` ภายใน The DAO กองทุนลงทุนแบบกระจายศูนย์ที่ระดมทุนเป็นอีเธอร์ได้กว่า 150 ล้านดอลลาร์จากผู้ร่วมลงทุนหลายพันคนในฤดูใบไม้ผลิปีนั้น ฟังก์ชันนี้ควรจะให้สมาชิกถอนส่วนแบ่งของตัวเองแล้วอัปเดตบันทึกภายในของกองทุน — แต่มันอัปเดตบันทึกหลังจาก ส่งเงินออกไปแล้ว ไม่ใช่ก่อน ผู้โจมตีจึงเรียกฟังก์ชันเดิมซ้ำจากภายใน

> ธุรกรรมเดียวกัน ก่อนที่ยอดคงเหลือจะถูกอัปเดต แล้วเรียกซ้ำวนเป็นลูปเรียกตัวเอง (recursive loop) ถี่ยิบ กว่าลูปจะหยุด อีเธอร์ราว 3.6 ล้านหน่วย — ราว 50 ล้านดอลลาร์ตามรายงานวันนั้น — ก็ไหลเข้าบัญชีที่ผู้โจมตีควบคุมอยู่ (Chainlink, "Reentrancy Attacks and The DAO Hack Explained") ไม่มีสิ่งใดที่ผู้โจมตีทำละเมิดโค้ดของสัญญาแม้แต่บรรทัดเดียว ทุกการเรียกใช้ตรงกับทุกกฎเป๊ะ

Ethereum's community split over what had just happened. One faction invoked the movement's own slogan, "code is law" — a contract's terms are whatever its bytecode executes, and a transaction that follows the code to the letter cannot be theft, whatever a plain-English reading of the fund's purpose would say. The other faction held that a contract has a purpose beyond its literal text, and on 20 July 2016, at block 1,920,000, the majority forked the entire Ethereum blockchain to reverse the theft — the holdout minority who kept the original chain became Ethereum Classic (The Block, "The DAO Hack at 10"). Both camps were looking at identical bytes of source code. They disagreed about what those bytes meant.

> ชุมชน Ethereum แตกออกเป็นสองฝ่ายเรื่องสิ่งที่เพิ่งเกิดขึ้น ฝ่ายหนึ่งอ้างคำขวัญของขบวนการเอง — "โค้ดคือกฎหมาย (code is law)" — เงื่อนไขของสัญญาคือสิ่งที่ bytecode ของมันสั่งให้ทำงานเท่านั้น และธุรกรรมที่ทำตามโค้ดตรงตัวเป๊ะย่อมไม่ใช่การขโมย ไม่ว่าการอ่านความมุ่งหมายของกองทุนแบบภาษาอังกฤษธรรมดาจะบอกว่าอย่างไรก็ตาม อีกฝ่ายเห็นว่าสัญญามีความมุ่งหมายที่อยู่เหนือตัวอักษรตามตัว และในวันที่ 20 กรกฎาคม 2016 ที่บล็อกที่ 1,920,000 เสียงข้างมากก็ฟอร์กบล็อกเชน Ethereum ทั้งสายเพื่อพลิกกลับการขโมยนั้น — ฝ่ายข้างน้อยที่ยืนกรานรักษาสายเดิมไว้กลายเป็น Ethereum Classic (The Block, "The DAO Hack at 10") ทั้งสองฝ่ายกำลังมองไบต์เดียวกันเป๊ะๆ ของซอร์สโค้ด พวกเขาไม่เห็นตรงกันว่าไบต์เหล่านั้นมีความหมายว่าอะไร

That disagreement is philosophy of language's oldest question, arriving through customs with a shipping label that says computer science. Fifty-five years earlier, Ludwig Wittgenstein had already named the trap waiting for anyone who thinks a written rule settles its own application: "This was our paradox: no course of action could be determined by a rule, because

any course of action can be made out to accord with the rule" (*Philosophical Investigations* §201, 1953). Saul Kripke's reading of the passage sharpens this into a skeptical challenge: nothing in a rule's text, and nothing in your past training on it, logically forces one continuation over any other of the infinitely many that "accord" with it (Kripke, *Wittgenstein on Rules and Private Language*, 1982). Computing built an entire discipline, formal semantics, on the bet that code was the one place this problem could finally be closed. The DAO is the place the bet came due: the reentrancy loop accorded perfectly with the contract's literal rules, and whether it violated *the rule* depended on an interpretation the bare text could not supply.

> ความเห็นไม่ตรงกันนั้นคือคำถามเก่าแก่ที่สุดของปรัชญาภาษา ที่เดินทางผ่านด่านศุลกากรพร้อมป้ายส่งของที่เขียนว่าวิทยาการคอมพิวเตอร์ (computer science) ห้าสิบห้าปีก่อนหน้านั้น Ludwig Wittgenstein ได้ตั้งชื่อกับดักที่รอคอยใครก็ตามที่คิดว่ากฎที่เขียนไว้กำหนดการใช้งานของตัวมันเองได้ไว้แล้วว่า "นี่คือปฏิทรรศน์ของเรา ไม่มีแนวทางปฏิบัติใดที่กฎจะกำหนดได้ เพราะแนวทางปฏิบัติใดก็ตามก็ทำให้ดูสอดคล้องกับกฎได้เสมอ" ("This was our paradox: no course of action could be determined by a rule, because any course of action can be made out to accord with the rule") (Philosophical Investigations §201, 1953) การอ่านข้อความนี้ของ Saul Kripke ลับให้มันคมขึ้นเป็นข้อท้าทายเชิงสงสัยนิยม — ไม่มีสิ่งใดในตัวบทของกฎ และไม่มีสิ่งใดในการฝึกฝนที่ผ่านมาของคุณกับมัน ที่บังคับทางตรรกะให้เลือกความต่อเนื่องแบบใดแบบหนึ่งเหนืออีกแบบหนึ่งจากบรรดาความเป็นไปได้นับไม่ถ้วนที่ "สอดคล้อง (accord)" กับมัน (Kripke, Wittgenstein on Rules and Private Language, 1982) วิทยาการคอมพิวเตอร์สร้างสาขาวิชาทั้งสาขาขึ้นมาเลย คือความหมายเชิงรูปนัย (formal semantics) โดยพนันว่าโค้ดคือที่แห่งเดียวที่ปัญหานี้จะถูกปิดได้จริงในที่สุด The DAO คือจุดที่การพนันนั้นถึงกำหนดต้องจ่าย — ลูปการเรียกซ้ำเข้าไปใหม่ (reentrancy) สอดคล้องกับกฎตามตัวอักษรของสัญญาอย่างสมบูรณ์แบบ และการที่มันละเมิด กฎ หรือไม่นั้น ขึ้นอยู่กับการตีความที่ตัวบทเปล่าๆ ให้ไม่ได้

Formal semantics offers two rival answers to "what does this program mean," and both were built to escape exactly Wittgenstein's problem by fixing meaning so completely that no ambiguity survives. Operational semantics, in the tradition Gordon Plotkin formalized in his 1981 Aarhus lecture notes "A Structural Approach to Operational Semantics" (Plotkin, DAIMI FN-19, Aarhus University, 1981), defines a program's meaning as the sequence of machine states it produces — meaning as a trace, checkable by simulation. Denotational semantics, which Christopher Strachey and Dana Scott built at Oxford's Programming Research Group, instead assigns each program a mathematical object directly, typically a function from inputs to outputs, and handles the hard case — a loop, a recursive call — by finding the least fixed point of that function over a structured space called a domain (Scott & Strachey, "Toward a Mathematical Semantics for Computer Languages," PRG-6, 1971). Both approaches genuinely succeed at their stated job: they make *what happens* fully determinate. Neither one touches Wittgenstein's actual question, which was never about the machine's next state — it was about whether the state the machine reaches is the one the rule was *for*. That gap is exactly where the Ethereum fork fell.

> ความหมายเชิงรูปนัย (formal semantics) เสนอคำตอบคู่แข่งกันสองแบบต่อคำถาม "โปรแกรมนี้มีความหมายว่าอะไร" และทั้งสองแบบถูกสร้างขึ้นมาเพื่อหนีปัญหาของ Wittgenstein โดยตรง ด้วยการตรึงความหมายให้แน่นจนไม่มีความกำกวมเหลือรอด ความหมายเชิงปฏิบัติการ (operational semantics) ตามประเพณีที่ Gordon Plotkin ทำให้เป็นรูปนัยไว้ในเอกสารประกอบการบรรยายที่ Aarhus ปี 1981 เรื่อง "A Structural Approach to Operational Semantics" (Plotkin, DAIMI FN-19, Aarhus University, 1981) นิยามความหมายของโปรแกรมว่าคือลำดับสถานะของเครื่องที่มันสร้างขึ้น — ความหมายในฐานะร่องรอย (trace) ที่ตรวจสอบได้ด้วยการจำลอง ส่วนความหมายเชิงนัยชี้ (denotational semantics) ซึ่ง Christopher Strachey และ Dana Scott สร้างขึ้นที่ Programming Research Group ของ Oxford กลับกำหนดวัตถุทางคณิตศาสตร์ให้กับแต่ละโปรแกรมโดยตรง โดยทั่วไปคือฟังก์ชันจากอินพุตไปเอาต์พุต และจัดการกรณียาก — ลูป การเรียกฟังก์ชันตัวเอง — ด้วยการหาจุดตรึงน้อยที่สุด (fixed point) ของฟังก์ชันนั้นเหนือพื้นที่ที่มีโครงสร้างเรียกว่าโดเมน (domain) (Scott & Strachey, "Toward a Mathematical

> Semantics for Computer Languages," PRG-6, 1971) ทั้งสองแนวทางทำสำเร็จตามที่ประกาศไว้จริง — มันทำให้ สิ่งที่เกิดขึ้น ชัดเจนแน่นอนเต็มที่ แต่ไม่มีแนวทางไหนแตะคำถามจริงของ Wittgenstein เลย ซึ่งไม่เคยเป็นเรื่องสถานะถัดไปของเครื่องตั้งแต่แรก — มันเป็นเรื่องว่าสถานะที่เครื่องไปถึงนั้นคือสถานะที่กฎมี ไว้เพื่อหรือเปล่าต่างหาก ช่องว่างนั้นแหละคือจุดที่การฟอร์ก Ethereum ตกลงไปพอดี

Protocols answer a different question about meaning: not what a message describes, but what it does. J. L. Austin's *How to Do Things with Words* — delivered as the William James Lectures at Harvard in 1955, published posthumously in 1962 after his death at 48 — named these performatives: utterances that are the act, not a report of it, valid only when "felicity conditions" hold, the right speaker, the right circumstance, the right procedure in the right order (SEP, "Speech Acts"). A commit does not describe a change to a repository; it is the change. A cryptographic signature does not report that you consented; producing it is the consenting. A TCP handshake does not describe an agreement to talk; completing it opens the channel. Protocol designers were enforcing felicity conditions mechanically — a malformed handshake simply fails, the way an insincere promise fails to bind — before Austin's vocabulary for it had even reached computing.

> โพรโทคอลตอบคำถามอีกแบบเกี่ยวกับความหมาย — ไม่ใช่ว่าข้อความหนึ่งบรรยายอะไร แต่มันทำอะไร J. L. Austin ใน How to Do Things with Words — บรรยายในชุด William James Lectures ที่ Harvard ปี 1955 ตีพิมพ์หลังเขาเสียชีวิตในปี 1962 ด้วยวัย 48 ปี — ตั้งชื่อสิ่งเหล่านี้ว่าประพจน์เชิงกระทำ (performative) คือถ้อยแถลงที่ตัวมันเองคือการกระทำ ไม่ใช่รายงานการกระทำ ใช้ได้ก็ต่อเมื่อ "เงื่อนไขความสำเร็จของวัจนกรรม (felicity condition)" เป็นจริง — ผู้พูดถูกคน สถานการณ์ถูกจังหวะ ขั้นตอนถูกลำดับ (SEP, "Speech Acts") การ commit ไม่ได้บรรยายการเปลี่ยนแปลงในที่เก็บ (repository) — มันคือการเปลี่ยนแปลงนั้นเอง ลายเซ็นเข้ารหัส (cryptographic signature) ไม่ได้รายงานว่าคุณยินยอม — การสร้างมันขึ้นมาคือการยินยอมนั้นเอง การจับมือ TCP (TCP handshake) ไม่ได้บรรยายข้อตกลงที่จะสื่อสารกัน — การทำมันจนสำเร็จคือการเปิดช่องสัญญาณ ผู้ออกแบบโพรโทคอลบังคับใช้เงื่อนไขความสำเร็จของวัจนกรรมแบบกลไกมาก่อนแล้ว — การจับมือที่ผิด

> รูปแบบก็แค่ล้มเหลวไปเฉยๆ แบบเดียวกับที่คำสัญญาที่ไม่จริงใจล้มเหลวในการผูกมัด — ตั้งแต่ก่อนที่คำศัพท์ของ Austin สำหรับเรื่องนี้จะมาถึงวงการคอมพิวเตอร์ด้วยซ้ำ

Meaning at the level of whole sentences split into two research programs computing has now run as a fifty-year experiment. Richard Montague's "The Proper Treatment of Quantification in Ordinary English" (1973, universally cited as PTQ) tried to give English the compositional rigor Frege had given arithmetic: every syntactic rule paired with a semantic rule, a sentence's meaning built strictly from its parts via intensional logic (SEP, "Montague Semantics"). Its rival needed no logical form at all. J. R. Firth's 1957 line — a word is known "by the company it keeps" — built on Zellig Harris's distributional hypothesis to found distributional semantics: a word's meaning just is the statistical shape of its neighboring words. Tomáš Mikolov and colleagues made that idea computationally cheap in 2013 with word2vec, and every large language model since has inherited the bet whole (Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," arXiv:1301.3781, 2013). For four decades computing funded Montague's structure-first program; the last decade switched almost entirely to Firth's context-first one — and the switch produced fluent machines built on a theory of meaning with no place in it for a proposition.

> ความหมายในระดับประโยคทั้งประโยคแตกออกเป็นสองสายโปรแกรมวิจัยที่วงการคอมพิวเตอร์ได้ทดลองมาแล้วห้าสิบปี งาน "The Proper Treatment of Quantification in Ordinary English" ของ Richard Montague (1973 ซึ่งทุกคนอ้างถึงในชื่อย่อ PTQ) พยายามมอบความเข้มงวดเชิงประกอบ (compositionality) ให้ภาษาอังกฤษแบบเดียวกับที่ Frege มอบให้เลขคณิต — ทุกกฎวากยสัมพันธ์จับคู่กับกฎความหมายหนึ่งข้อ ความหมายของประโยคถูกสร้างขึ้นอย่างเคร่งครัดจากส่วนประกอบของมันผ่านตรรกะเชิงเจตนา (intensional logic) (SEP, "Montague Semantics") คู่แข่งของมันไม่ต้องการรูปแบบตรรกะ (logical form) เลยแม้แต่น้อย ประโยคของ J. R. Firth ในปี 1957 ที่ว่าคำหนึ่งเป็นที่รู้จัก "จากพวกที่มันคบหา" ต่อยอดจากสมมติฐานเชิงการกระจาย (distributional hypothesis) ของ Zellig Harris จนตั้งเป็นความหมายเชิงการกระจาย (distributional semantics) —

> ความหมายของคำหนึ่งก็คือรูปทรงเชิงสถิติของคำที่อยู่ข้างเคียงมันเท่านั้นเอง Tomáš Mikolov และคณะทำให้ไอเดียนี้คำนวณได้ถูกลงในปี 2013 ด้วย word2vec และโมเดลภาษาขนาดใหญ่ (large language model) ทุกตัวนับแต่นั้นก็รับพนันเดิมพันนี้ไว้เต็มๆ (Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," arXiv:1301.3781, 2013) ตลอดสี่ทศวรรษวงการคอมพิวเตอร์ทุ่มทุนให้โปรแกรมแบบโครงสร้างก่อนของ Montague ทศวรรษล่าสุดสลับไปเกือบทั้งหมดยังโปรแกรมแบบบริบทก่อนของ Firth — และการสลับนั้นผลิตเครื่องจักรที่พูดคล่องแคล่วขึ้นมาบนทฤษฎีความหมายที่ไม่มีที่ว่างให้กับประพจน์ (proposition) เลยสักนิด

Which is the test an LLM actually runs on philosophy of language, and the verdict splits. Wittgenstein's later view — "the meaning of a word is its use in the language" (*Philosophical Investigations* §43) — looks vindicated by a model that has never touched a referent and still produces sentences no fluent speaker would flag as wrong. But Stevan Harnad named the missing piece in 1990, before any of this was built: a system that manipulates symbols by their shapes and their relations to other symbols alone, never anchored to what the symbols are about, is stuck trying to learn Chinese from a Chinese–Chinese dictionary — perfect internal coherence, zero contact with the world (Harnad, "The Symbol Grounding Problem," *Physica D* 42, 1990). The large language model is the best evidence a pure use-theory of meaning has ever been handed, and the clearest working model of exactly what grounding critics warned an ungrounded use-theory would look like.

> นี่แหละคือบททดสอบที่โมเดลภาษาขนาดใหญ่ (LLM) รันจริงต่อปรัชญาภาษา และคำตัดสินก็แตกออกเป็นสองทาง มุมมองยุคหลังของ Wittgenstein — "ความหมายของคำหนึ่งคือการใช้งานของมันในภาษา" (Philosophical Investigations §43) — ดูเหมือนจะได้รับการพิสูจน์ว่าถูกต้องจากโมเดลที่ไม่เคยแตะสิ่งอ้างอิง (referent) เลยสักครั้ง แต่ก็ยังผลิตประโยคที่ไม่มีเจ้าของภาษาคนไหนทักท้วงว่าผิด แต่ Stevan Harnad ได้ตั้งชื่อชิ้นส่วนที่ขาดหายไปตั้งแต่ปี 1990 ก่อนที่สิ่งเหล่านี้จะถูกสร้างขึ้นมาด้วยซ้ำ — ระบบที่จัดการสัญลักษณ์ผ่านรูปร่างของมันและความสัมพันธ์กับสัญลักษณ์อื่นเพียงอย่างเดียว โดยไม่เคยยึดโยงกับสิ่งที่สัญลักษณ์เหล่านั้นพูดถึง

> จริงๆ ก็เหมือนติดอยู่กับการพยายามเรียนภาษาจีนจากพจนานุกรมจีน–จีน — สอดคล้องกันในตัวเองอย่างสมบูรณ์แบบ แต่ไม่แตะโลกจริงเลยแม้แต่น้อย (Harnad, "The Symbol Grounding Problem," Physica D 42, 1990) โมเดลภาษาขนาดใหญ่คือหลักฐานที่ดีที่สุดเท่าที่ทฤษฎีความหมายจากการใช้ (use theory of meaning) แบบบริสุทธิ์เคยได้รับมา และเป็นแบบจำลองใช้งานได้ที่ชัดเจนที่สุดของสิ่งที่นักวิจารณ์เรื่องการยึดโยง (grounding) เคยเตือนไว้ว่าทฤษฎีความหมายจากการใช้ที่ไม่มีการยึดโยงจะมีหน้าตาเป็นอย่างไร

Connections · เชื่อมโยง

Chapter 01 supplies the logical apparatus — Frege's quantifiers — that both Montague and denotational semantics depend on. Chapter 02 returns to the type/token question this chapter only brushes: is a smart contract's deployed bytecode the same thing as its Solidity source? Chapter 04 runs the same grounding argument through the Chinese Room, and chapter 13 tests it against machines trained on nothing but text. Chapter 12 shows what happens when "the code says what it means" is pushed furthest — a proof assistant, where the rules really do settle every case.

> บทที่ 01 ให้กลไกเชิงตรรกะ — ตัวบ่งปริมาณ (quantifier) ของ Frege — ที่ทั้ง Montague และความหมายเชิงนัยชี้ (denotational semantics) ต้องพึ่งพา บทที่ 02 กลับไปหาปัญหาชนิด/ตัวอย่างจริง (type/token) ที่บทนี้แตะเพียงผิวเผิน — bytecode ที่ deploy จริงของ smart contract คือสิ่งเดียวกับซอร์ส Solidity ของมันหรือเปล่า บทที่ 04 ใช้ข้อโต้แย้งเรื่องการยึดโยง (grounding) เดียวกันนี้ผ่านห้องจีน (Chinese Room) และบทที่ 13 ทดสอบมันกับเครื่องจักรที่ฝึกมาจากข้อความล้วนๆ บทที่ 12 แสดงให้เห็นว่าเกิดอะไรขึ้นเมื่อ "โค้ดบอกความหมายของตัวมันเอง" ถูกผลักไปสุดทาง — proof assistant ที่ซึ่งกฎเป็นผู้ตัดสินทุกกรณีจริงๆ

Open questions · คำถามที่ยังเปิดอยู่

If formal semantics can specify a program's behavior completely and still leave open what the program was *for*, has it solved Wittgenstein's problem, or only changed which question gets answered? Did distributional se-

mantics win because it is a better theory of meaning, or because it is the theory computers could finally afford to run at scale? And if an ungrounded system passes every behavioral test of understanding, what work is "grounding" still doing, beyond naming a discomfort?

> ถ้าความหมายเชิงรูปนัย (formal semantics) ระบุพฤติกรรมของโปรแกรมได้ครบถ้วนสมบูรณ์ แต่ยังทิ้งคำถามไว้ว่าโปรแกรมนั้นมี ไว้เพื่อ อะไร มันแก้ปัญหาของ Wittgenstein ได้จริงหรือแค่เปลี่ยนคำถามที่ถูกตอบ ความหมายเชิงการกระจาย (distributional semantics) ชนะเพราะมันเป็นทฤษฎีความหมายที่ดีกว่า หรือเพราะมันเป็นทฤษฎีเดียวที่คอมพิวเตอร์รันได้ไหวในที่สุดเมื่อขยายสเกล และถ้าระบบที่ไม่มีการยึดโยงผ่านทุกบททดสอบเชิงพฤติกรรมของความเข้าใจได้ "การยึดโยง (grounding)" ยังทำหน้าที่อะไรอยู่ นอกจากตั้งชื่อให้กับความรู้สึกไม่สบายใจ

Go deeper · อ่านต่อ

- Ludwig Wittgenstein, *Philosophical Investigations*, trans. G. E. M. Anscombe, Blackwell, 1953 — §§138–242 is the rule-following argument this chapter borrows.

  > Ludwig Wittgenstein, Philosophical Investigations, trans. G. E. M. Anscombe, Blackwell, 1953 — §§138–242 คือข้อโต้แย้งเรื่องการทำตามกฎ (rule-following) ที่บทนี้หยิบยืมมาใช้

- Richard Montague, "The Proper Treatment of Quantification in Ordinary English," in *Approaches to Natural Language*, ed. Hintikka, Moravcsik & Suppes, Reidel, 1973, pp. 221–242 (PTQ), read against Tomáš Mikolov, Kai Chen, Greg Corrado & Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv:1301.3781, 2013 — the two theories of meaning computing actually tried, forty years apart.

> Richard Montague, "The Proper Treatment of Quantification in Ordinary English," in Approaches to Natural Language, ed. Hintikka, Moravcsik & Suppes, Reidel, 1973, pp. 221–242 (PTQ), read against Tomáš Mikolov, Kai Chen, Greg Corrado & Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv:1301.3781, 2013 — สองทฤษฎีความหมายที่วงการคอมพิวเตอร์ลองใช้จริง ห่างกันสี่สิบปี

- Stanford Encyclopedia of Philosophy, "Montague Semantics" — the standing account of where formal and natural-language semantics meet.

> Stanford Encyclopedia of Philosophy, "Montague Semantics" — บันทึกหลักเรื่องจุดที่ความหมายเชิงรูปนัยกับความหมายภาษาธรรมชาติมาบรรจบกัน

CHAPTER 06

Ethics

จริยศาสตร์

This chapter sits in Part I, the Court of Philosophy — the sixth hearing, where philosophy asks the machine whether a decision keeps its moral weight once a system makes it.

บทนี้อยู่ในภาค I ศาลปรัชญา — การไต่สวนครั้งที่ 6 ที่ปรัชญาตั้งคำถามกับเครื่องจักรว่าการตัดสินใจหนึ่งยังคงน้ำหนักเชิงศีลธรรมของมันไว้หรือไม่ เมื่อระบบเป็นผู้ตัดสินใจแทน

Can a decision be delegated without losing its morality?

การตัดสินใจหนึ่งถูกส่งต่อให้ผู้อื่นทำแทนได้โดยไม่สูญเสียมิติทางศีลธรรมของมันหรือไม่

On a spring afternoon in 2014, eighteen-year-old Brisha Borden and a friend grabbed an unlocked kid's bike and scooter in a Fort Lauderdale suburb, rode them half a block, and were arrested for burglary and petty theft — total value, $80. The previous summer, forty-one-year-old Vernon Prater had been picked up for shoplifting $86.35 of tools from a Home Depot; he had already been convicted of two armed robberies and an attempted armed robbery. Both were scored by COMPAS, a proprietary risk-assessment tool used across Broward County, Florida, to predict the likelihood of reoffending. Borden, Black and a first-time adult offender, scored 8 out of 10 — high risk. Prater, white and a repeat violent felon, scored 3 — low risk. Two years later, ProPublica's investigation found the algorithm had gotten it backward: Borden was not charged with any further crime; Prater was soon incarcerated again (Angwin et al., "Machine Bias," ProPublica, 23 May 2016).

Across more than 7,000 Broward County cases, Black defendants who did not reoffend were misclassified as high-risk at roughly twice the rate of white defendants who did not reoffend.

> บ่ายวันหนึ่งในฤดูใบไม้ผลิปี 2014 Brisha Borden วัยสิบแปดปีกับเพื่อนคว้ารถจักรยานและสกู๊ตเตอร์เด็กที่ไม่ได้ล็อกในย่านชานเมือง Fort Lauderdale ขี่ไปได้ครึ่งช่วงตึกก็ถูกจับข้อหางัดแงะและลักทรัพย์เล็กน้อย — มูลค่ารวม 80 ดอลลาร์ ฤดูร้อนก่อนหน้านั้น Vernon Prater วัยสี่สิบเอ็ดปีถูกจับข้อหาขโมยเครื่องมือมูลค่า 86.35 ดอลลาร์จาก Home Depot เขาเคยถูกตัดสินว่ามีความผิดฐานปล้นทรัพย์ด้วยอาวุธมาแล้วสองครั้งและพยายามปล้นทรัพย์ด้วยอาวุธอีกหนึ่งครั้ง ทั้งคู่ถูกให้คะแนนโดย COMPAS เครื่องมือประเมินความเสี่ยงกรรมสิทธิ์เอกชนที่ใช้ทั่ว Broward County รัฐฟลอริดา เพื่อทำนายโอกาสที่จะกระทำผิดซ้ำ Borden ซึ่งเป็นคนผิวดำและกระทำผิดในฐานะผู้ใหญ่เป็นครั้งแรก ได้คะแนน 8 จาก 10 — ความเสี่ยงสูง Prater ซึ่งเป็นคนผิวขาวและเป็นอาชญากรรุนแรงซ้ำซาก ได้คะแนน 3 — ความเสี่ยงต่ำ สองปีต่อมา การสืบสวนของ ProPublica พบว่าอัลกอริทึมทำนายกลับด้านกันเลย — Borden ไม่ถูกตั้งข้อหาก่ออาชญากรรมเพิ่มเติมอีกเลย ส่วน Prater กลับเข้าคุกอีกครั้งในเวลาไม่นาน (Angwin et al., "Machine Bias," ProPublica, 23 May 2016) จากคดีในเขต Broward County กว่า 7,000 คดี จำเลยผิวดำที่ไม่ได้กระทำผิดซ้ำถูกจัดว่าเป็นความเสี่ยงสูงผิดพลาดในอัตราสูงกว่าจำเลยผิวขาวที่ไม่ได้กระทำผิดซ้ำเกือบสองเท่า

Northpointe, COMPAS's maker, did not deny the numbers — it denied that they proved bias. Its scores were *calibrated*: among defendants who scored, say, 7, roughly the same share reoffended whether they were Black or white. ProPublica had measured a different quantity — the *error rates* within each racial group — and by that measure the tool was plainly unequal. Both sides were right about their own arithmetic, and this is where the story stops being a scandal about one company's software and becomes a theorem. Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan, and independently Alexandra Chouldechova, proved in 2016–17 that when the base rate of the outcome differs between two groups — as recidivism rates did between Black and white defendants in this data — no risk score can generally satisfy both calibration and equal error rates at once, except in narrow special cases (Kleinberg, Mullainathan & Raghavan, "Inherent Trade-Offs in the Fair

Determination of Risk Scores," arXiv:1609.05807; Chouldechova, "Fair Prediction with Disparate Impact," *Big Data* 5(2), 2017). COMPAS and ProPublica were not arguing about facts. They were standing on opposite sides of an impossibility result and each calling the other's axiom the wrong one — which means "make the algorithm fair" is not an engineering instruction until someone first answers a question philosophy has never closed: fair by which of several mutually exclusive criteria, and who gets to choose.

> Northpointe ผู้สร้าง COMPAS ไม่ได้ปฏิเสธตัวเลขเหล่านั้น — แต่ปฏิเสธว่ามันพิสูจน์อคติ คะแนนของบริษัทถูก ปรับเทียบ (calibration) แล้ว — ในบรรดาจำเลยที่ได้คะแนน สมมติว่า 7 สัดส่วนที่กระทำผิดซ้ำใกล้เคียงกันไม่ว่าจะเป็นคนผิวดำหรือผิวขาว ส่วน ProPublica วัดปริมาณอีกแบบหนึ่ง — อัตราความผิดพลาด (error rate) ภายในแต่ละกลุ่มเชื้อชาติ — และเมื่อวัดแบบนั้น เครื่องมือนี้ไม่เท่าเทียมกันอย่างชัดเจน ทั้งสองฝ่ายถูกต้องในเลขคณิตของตัวเอง และตรงนี้แหละที่เรื่องราวเลิกเป็นเรื่องอื้อฉาวของซอฟต์แวร์บริษัทเดียว แล้วกลายเป็นทฤษฎีบท Jon Kleinberg, Sendhil Mullainathan และ Manish Raghavan และแยกกันโดย Alexandra Chouldechova พิสูจน์ในปี 2016–17 ว่าเมื่ออัตราฐาน (base rate) ของผลลัพธ์แตกต่างกันระหว่างสองกลุ่ม — ดังที่อัตราการกระทำผิดซ้ำแตกต่างกันระหว่างจำเลยผิวดำกับผิวขาวในข้อมูลชุดนี้ — ไม่มีคะแนนความเสี่ยงใดที่จะสนองทั้งการปรับเทียบและอัตราความผิดพลาดที่เท่ากันได้พร้อมกันโดยทั่วไป ยกเว้นในกรณีพิเศษแคบๆ (Kleinberg, Mullainathan & Raghavan, "Inherent Trade-Offs in the Fair Determination of Risk Scores," arXiv:1609.05807; Chouldechova, "Fair Prediction with Disparate Impact," Big Data 5(2), 2017) COMPAS กับ ProPublica ไม่ได้เถียงกันเรื่องข้อเท็จจริง พวกเขายืนอยู่คนละฝั่งของผลลัพธ์เชิงความเป็นไปไม่ได้ (impossibility result) แล้วต่างฝ่ายต่างบอกว่าสัจพจน์ของอีกฝ่ายผิด — ซึ่งหมายความว่า "ทำให้อัลกอริทึมเป็นธรรม" ไม่ใช่คำสั่งเชิงวิศวกรรมเลย จนกว่าจะมีใครตอบคำถามที่ปรัชญาไม่เคยปิดได้ก่อน — เป็นธรรมตามเกณฑ์ไหนในบรรดาเกณฑ์หลายข้อที่ขัดแย้งกันเอง และใครเป็นผู้เลือก

Delegating the decision does not just relocate the fairness question — it can dissolve the chain of responsibility that used to answer for it. A system that adapts from data after deployment can produce an output its own engin-

eers cannot predict or fully reconstruct, which breaks the ordinary assumption that a designer's intent traces forward into a machine's behavior. Andreas Matthias named this the *responsibility gap*: when a learning automaton's actions no longer trace predictably to any human's decision, neither the manufacturer nor the operator can be justly held liable by the old rules, and yet someone has been harmed (Matthias, "The Responsibility Gap," *Ethics and Information Technology* 6(3), 2004). Helen Nissenbaum had already catalogued the mechanics of this a decade earlier — "many hands" spreading a single harm across enough contributors that no one holds the whole of it, bugs treated as acts of nature, blame displaced onto "the computer," and software shipped under licenses that disclaim liability by default (Nissenbaum, "Computing and Accountability," *CACM* 37(1), 1994). A gap in responsibility is not an absence of harm. It is harm with the addressee missing.

> การมอบหมายการตัดสินใจไม่ได้แค่ย้ายคำถามเรื่องความเป็นธรรมไปที่อื่น — มันสามารถละลายห่วงโซ่ความรับผิดชอบที่เคยตอบคำถามนั้นได้ทั้งหมด ระบบที่ปรับตัวจากข้อมูลหลังถูกนำไปใช้งานจริงสามารถผลิตผลลัพธ์ที่วิศวกรผู้สร้างเองก็ทำนายหรือสร้างย้อนกลับให้ครบไม่ได้ ซึ่งทำลายสมมติฐานปกติที่ว่าเจตนาของผู้ออกแบบจะสืบต่อไปถึงพฤติกรรมของเครื่องจักรได้เสมอ Andreas Matthias ตั้งชื่อสิ่งนี้ว่า ช่องว่างความรับผิดชอบ (responsibility gap) — เมื่อการกระทำของออโตมาตาที่เรียนรู้ได้ไม่สืบย้อนไปหาการตัดสินใจของมนุษย์คนใดได้อย่างคาดเดาได้อีกต่อไป ทั้งผู้ผลิตและผู้ควบคุมต่างก็ไม่อาจถูกให้รับผิดอย่างเป็นธรรมตามกฎเดิมได้ ทั้งที่มีคนได้รับความเสียหายจริง (Matthias, "The Responsibility Gap," Ethics and Information Technology 6(3), 2004) Helen Nissenbaum ได้ไล่เรียงกลไกของเรื่องนี้ไว้แล้วสิบปีก่อนหน้า — "มือหลายมือ (many hands)" กระจายความเสียหายเดียวออกไปยังผู้มีส่วนร่วมมากพอจนไม่มีใครแบกรับทั้งหมดไว้คนเดียว บั๊กถูกมองเป็นภัยธรรมชาติ ความผิดถูกโยนไปให้ "คอมพิวเตอร์" และซอฟต์แวร์ถูกส่งมอบภายใต้สัญญาอนุญาตที่ปฏิเสธความรับผิดเป็นค่าตั้งต้น (Nissenbaum, "Computing and Accountability," CACM 37(1), 1994) ช่องว่างในความรับผิดชอบไม่ใช่การไม่มีความเสียหาย มันคือความเสียหายที่ไม่มีผู้รับผิดชอบให้ส่งถึง

If responsibility can slip away from a system, philosophy has to ask what a system *is*, morally — object, agent, or something newly in between. Luciano Floridi and Jeff Sanders argued that "moral agent" need not require free will, consciousness, or intention at all: an entity that is interactive (responds to stimuli by changing state), autonomous (can change state without external stimulus), and adaptable (can change the very rules by which it changes state) qualifies as a moral agent at the relevant level of abstraction, whatever else it lacks (Floridi & Sanders, "On the Morality of Artificial Agents," *Minds and Machines* 14(3), 2004). That loosens moral agency enough to fit a recommendation engine or a trading bot without claiming it has an inner life. It leaves untouched, and does not answer, the harder question now being asked of the systems this book keeps discussing: whether something interactive and adaptive enough could also become a moral *patient* — able to be wronged, not just to act — a question no consensus method yet resolves.

> ถ้าความรับผิดชอบสามารถหลุดลอยไปจากระบบได้ ปรัชญาก็ต้องถามว่าระบบหนึ่งคืออะไร ในเชิงศีลธรรม — วัตถุ ผู้กระทำ หรือสิ่งใหม่ที่อยู่ตรงกลาง Luciano Floridi และ Jeff Sanders โต้แย้งว่า "ผู้กระทำเชิงศีลธรรม (moral agent)" ไม่จำเป็นต้องมีเจตจำนงเสรี จิตสำนึก หรือเจตนาเลยแม้แต่น้อย — สิ่งใดก็ตามที่มีปฏิสัมพันธ์ (ตอบสนองต่อสิ่งเร้าด้วยการเปลี่ยนสถานะ) เป็นอิสระ (เปลี่ยนสถานะได้โดยไม่ต้องมีสิ่งเร้าจากภายนอก) และปรับตัวได้ (เปลี่ยนกฎที่ใช้เปลี่ยนสถานะของตัวมันเองได้) ก็มีคุณสมบัติเป็นผู้กระทำเชิงศีลธรรมในระดับ abstraction ที่เกี่ยวข้อง ไม่ว่ามันจะขาดคุณสมบัติอื่นใดก็ตาม (Floridi & Sanders, "On the Morality of Artificial Agents," Minds and Machines 14(3), 2004) นั่นผ่อนคลายเงื่อนไขของการเป็นผู้กระทำเชิงศีลธรรมมากพอที่จะครอบคลุมเอนจินแนะนำ (recommendation engine) หรือบอตเทรดหุ้น โดยไม่ต้องอ้างว่ามันมีชีวิตภายใน แต่มันไม่แตะและไม่ตอบคำถามที่ยากกว่า ซึ่งกำลังถูกถามกับระบบที่หนังสือเล่มนี้พูดถึงซ้ำแล้วซ้ำเล่า — สิ่งที่มีปฏิสัมพันธ์และปรับตัวได้มากพอจะกลายเป็น ผู้ถูกกระทำเชิงศีลธรรม (moral patient) ได้ด้วยหรือไม่ — คือถูกกระทำผิดได้ ไม่ใช่แค่กระทำได้ — คำถามที่ยังไม่มีวิธีตกลงกันได้เป็นเอกฉันท์

Alignment research turns out to be metaethics wearing an engineer's badge. Iason Gabriel's 2020 analysis distinguishes several things "align an AI with human values" could mean — instructions, revealed preferences, idealized preferences, interests, or values — and argues the central difficulty is not discovering the one true set of moral principles, but designing principles fair enough to win reflective endorsement across people who deeply and permanently disagree about ethics (Gabriel, "Artificial Intelligence, Values, and Alignment," *Minds and Machines* 30, 2020). That is exactly the problem political philosophers call the problem of legitimacy under pluralism — alignment did not invent a new question; it gave an old one a training loop and a deadline.

> งานวิจัยด้าน alignment กลายเป็นเมทาเอทิกส์ (metaethics) ที่สวมป้ายวิศวกรอยู่ดี บทวิเคราะห์ของ Iason Gabriel ในปี 2020 แยกแยะสิ่งต่างๆ ที่คำว่า "ปรับ AI ให้สอดคล้องกับคุณค่าของมนุษย์ (align an AI with human values)" อาจหมายถึง — คำสั่ง ความพึงพอใจที่แสดงออก ความพึงพอใจในอุดมคติ ผลประโยชน์ หรือคุณค่า — และโต้แย้งว่าความยากหลักไม่ใช่การค้นหาชุดหลักศีลธรรมที่แท้จริงเพียงชุดเดียว แต่คือการออกแบบหลักการที่เป็นธรรมมากพอจะได้รับการรับรองอย่างใคร่ครวญจากผู้คนที่เห็นต่างกันอย่างลึกซึ้งและถาวรเรื่องจริยธรรม (Gabriel, "Artificial Intelligence, Values, and Alignment," Minds and Machines 30, 2020) นั่นคือปัญหาเดียวกันเป๊ะกับที่นักปรัชญาการเมืองเรียกว่าปัญหาความชอบธรรมภายใต้พหุนิยม (legitimacy under pluralism) — alignment ไม่ได้คิดค้นคำถามใหม่ขึ้นมา มันแค่เอาลูปการเทรน (training loop) กับเดดไลน์ไปใส่ให้คำถามเก่า

Privacy submits to the same delegation problem in a quieter register. Helen Nissenbaum's *contextual integrity*, developed across the 1990s and set out fully in *Privacy in Context* (2010), argues that privacy is neither secrecy nor control over information but the expectation that information flows according to the norms proper to the context it came from: a diagnosis flowing from doctor to insurer for billing is a different flow, governed by different norms, than the same fact flowing from doctor to employer (Nissenbaum, *Privacy in Context*, Stanford UP, 2010). Most data-collection systems are context-blind by construction — a field in a database carries no memory

of which norms it was gathered under — which is why so much that feels like a privacy violation involves no secret at all, only information moved into the wrong context.

> ความเป็นส่วนตัวเผชิญปัญหาการมอบหมายแบบเดียวกันนี้ในโทนที่เงียบกว่า ความครบถ้วนเชิงบริบท (contextual integrity) ของ Helen Nissenbaum ซึ่งพัฒนาขึ้นตลอดทศวรรษ 1990 และวางกรอบไว้เต็มรูปแบบใน Privacy in Context (2010) โต้แย้งว่าความเป็นส่วนตัวไม่ใช่ทั้งความลับและไม่ใช่การควบคุมข้อมูล แต่คือความคาดหวังว่าข้อมูลจะไหลไปตามบรรทัดฐานที่เหมาะสมกับบริบทที่มันมาจาก — การวินิจฉัยโรคที่ไหลจากหมอไปยังบริษัทประกันเพื่อเรียกเก็บเงินคือการไหลแบบหนึ่งถูกกำกับด้วยบรรทัดฐานแบบหนึ่ง ต่างจากข้อเท็จจริงเดียวกันที่ไหลจากหมอไปยังนายจ้าง (Nissenbaum, Privacy in Context, Stanford UP, 2010) ระบบเก็บข้อมูลส่วนใหญ่ตาบอดต่อบริบทโดยโครงสร้างของมันเอง — ฟิลด์หนึ่งในฐานข้อมูลไม่มีความทรงจำว่ามันถูกเก็บมาภายใต้บรรทัดฐานใด — นี่คือเหตุผลที่สิ่งซึ่งให้ความรู้สึกเหมือนการละเมิดความเป็นส่วนตัวจำนวนมากไม่มีความลับใดๆ เกี่ยวข้องเลย มีแต่ข้อมูลที่ถูกย้ายเข้าไปในบริบทที่ผิด

Google convened an eight-member external AI ethics council, ATEAC, on 26 March 2019 and dissolved it entirely nine days later, after employee protest over two of its members made the board's continued existence untenable (MIT Technology Review, "Google Has Now Cancelled Its AI Ethics Board," 5 April 2019). An advisory board's authority depends entirely on the goodwill of the institution that convened it and can be withdrawn as easily as it was granted. Mechanism design does not ask for goodwill: a fairness constraint written into a training objective, or a differential-privacy bound written into a query system, keeps operating on the day nobody is in the room to enforce it — which is a narrower kind of ethics than a board's, but a harder one to dissolve by memo.

> Google ตั้งสภาจริยธรรม AI ภายนอกแปดคนชื่อ ATEAC ขึ้นในวันที่ 26 มีนาคม 2019 แล้วยุบทิ้งทั้งหมดในอีกเก้าวันต่อมา หลังพนักงานประท้วงกรรมการสองคนในนั้นจนทำให้สภาอยู่ต่อไม่ได้ (MIT Technology Review, "Google Has Now Cancelled Its AI Ethics Board," 5 April 2019) อำนาจของบอร์ดที่ปรึกษาขึ้นอยู่กับ

> ความเต็มใจของสถาบันที่ตั้งมันขึ้นมาทั้งหมด และถอนออกได้ง่ายพอๆ กับที่มอบให้มา mechanism design (การออกแบบกลไก) ไม่ต้องพึ่งความเต็มใจ — ข้อจำกัดความเป็นธรรมที่เขียนไว้ในเป้าหมายการเทรน หรือขอบเขต differential privacy ที่เขียนไว้ในระบบสืบค้น ยังทำงานต่อไปได้แม้ในวันที่ไม่มีใครอยู่ในห้องคอยบังคับใช้มัน — ซึ่งเป็นจริยธรรมประเภทที่แคบกว่าของบอร์ด แต่ยุบทิ้งด้วยบันทึกภายในฉบับเดียวได้ยากกว่ามาก

Connections · เชื่อมโยง

Chapter 02 supplies the deeper question of what kind of thing a "decision" made by software even is. Chapter 07 takes platform power into the register of governance, where the question stops being who is accountable and becomes who is sovereign. Chapter 13 returns to the responsibility gap through interpretability — can you hold a system accountable for a decision no one, including its designers, can fully read back out of it?

> บทที่ 02 ให้คำถามที่ลึกกว่านั้นว่า "การตัดสินใจ" ที่ทำโดยซอฟต์แวร์คือสิ่งประเภทไหนกันแน่ บทที่ 07 พาอำนาจของแพลตฟอร์มเข้าสู่ทะเบียนของธรรมาภิบาล ที่ซึ่งคำถามเลิกเป็นเรื่องใครรับผิดชอบ แล้วกลายเป็นใครมีอธิปไตย บทที่ 13 กลับมาที่ช่องว่างความรับผิดชอบผ่าน interpretability — คุณให้ระบบรับผิดชอบต่อการตัดสินใจที่ไม่มีใคร รวมถึงผู้ออกแบบมันเอง อ่านย้อนออกมาได้ครบถ้วนได้หรือไม่

Open questions · คำถามที่ยังเปิดอยู่

If calibration and error-rate balance cannot both hold, is choosing between them a technical decision at all, or is every deployed risk score a silent vote on which kind of injustice a society will tolerate? Does Floridi and Sanders's minimal moral agency dissolve responsibility further by making blame diffusible to "the system," or does it at least make the diffusion visible enough

to argue about? And if legitimacy under pluralism was always the real problem behind alignment, what does computer science have to offer political philosophy that centuries of contract theory did not already try?

> ถ้าการปรับเทียบ (calibration) กับความสมดุลของอัตราความผิดพลาดเป็นจริงพร้อมกันไม่ได้ การเลือกระหว่างสองสิ่งนี้ยังนับเป็นการตัดสินใจเชิงเทคนิคอยู่หรือไม่ หรือคะแนนความเสี่ยงทุกตัวที่ถูกนำไปใช้จริงคือการโหวตเงียบๆ ว่าสังคมจะยอมทนความอยุติธรรมแบบไหน ผู้กระทำเชิงศีลธรรมแบบขั้นต่ำของ Floridi และ Sanders ทำให้ความรับผิดชอบละลายไปมากขึ้นอีกด้วยการทำให้ความผิดกระจายไปที่ "ระบบ" ได้ หรืออย่างน้อยมันก็ทำให้การกระจายนั้นมองเห็นได้ชัดพอจะถกเถียงกันได้ และถ้าความชอบธรรมภายใต้พหุนิยม (legitimacy under pluralism) คือปัญหาจริงที่อยู่เบื้องหลัง alignment มาตลอด วิทยาการคอมพิวเตอร์มีอะไรจะเสนอให้ปรัชญาการเมืองที่ทฤษฎีสัญญาประชาคมนับศตวรรษที่ผ่านมายังไม่เคยลองมาก่อนบ้าง

Go deeper · อ่านต่อ

— Cathy O'Neil, *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*, Crown, 2016 — the general-audience case this chapter's theorem underwrites.

> Cathy O'Neil, Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy, Crown, 2016 — กรณีศึกษาสำหรับผู้อ่านทั่วไปที่ทฤษฎีบทของบทนี้ค้ำอยู่เบื้องหลัง

— Jon Kleinberg, Sendhil Mullainathan & Manish Raghavan, "Inherent Trade-Offs in the Fair Determination of Risk Scores," arXiv:1609.05807, 2016, read against Alexandra Chouldechova, "Fair Prediction with Disparate Impact," *Big Data* 5(2), 2017 — the impossibility result from both directions.

> Jon Kleinberg, Sendhil Mullainathan & Manish Raghavan, "Inherent Trade-Offs in the Fair Determination of Risk Scores," arXiv:1609.05807, 2016, read against Alexandra Chouldechova, "Fair Prediction with Disparate Impact," Big Data 5(2), 2017 — ผลลัพธ์เชิงความเป็นไปไม่ได้ (impossibility result) จากทั้งสองทิศทาง

— Stanford Encyclopedia of Philosophy, "Ethics of Artificial Intelligence and Robotics" — the standing map of this chapter's whole terrain.

> Stanford Encyclopedia of Philosophy, "Ethics of Artificial Intelligence and Robotics" — แผนที่หลักของภูมิประเทศทั้งหมดในบทนี้

CHAPTER 07

Political Philosophy

ปรัชญาการเมือง

This chapter sits in Part I, the Court of Philosophy — the seventh hearing, where philosophy asks the machine who is actually governing when its code sets the terms.

บทนี้อยู่ในภาค I ศาลปรัชญา — การไต่สวนครั้งที่ 7 ที่ปรัชญาตั้งคำถามกับเครื่องจักรว่าใครกันแน่ที่ปกครองอยู่ เมื่อโค้ดของมันเป็นผู้กำหนดเงื่อนไข

Who governs when code is law?

ใครปกครอง เมื่อโค้ดคือกฎหมาย

On 16 August 2017, days after the violence in Charlottesville, Cloudflare CEO Matthew Prince terminated the account of The Daily Stormer, a neo-Nazi website, cutting off the DNS and content-delivery service that kept it reachable on the open internet. In an internal email later published by Gizmodo, Prince described his own reasoning with unusual honesty: "Literally, I woke up in a bad mood and decided someone shouldn't be allowed on the Internet," he wrote. "No one should have that power" (Gizmodo, "Cloudflare CEO on Terminating Service to Neo-Nazi Site," 2017). No court had ordered the takedown. No legislature had voted on it. One executive, at one infrastructure company, flipped a switch, and a publication that had survived years of public outrage disappeared from the reachable internet within hours. Prince was right that no one should have that power — and right, too, that in that moment, he had it.

> วันที่ 16 สิงหาคม 2017 ไม่กี่วันหลังความรุนแรงที่ Charlottesville Matthew Prince ซีอีโอของ Cloudflare ยกเลิกบัญชีของ The Daily Stormer เว็บไซต์นีโอนาซี ตัดบริการ DNS และ content-delivery ที่ทำให้เว็บนี้เข้าถึงได้บนอินเทอร์เน็ตสาธารณะ ในอีเมลภายในที่ Gizmodo เผยแพร่ในภายหลัง Prince อธิบายเหตุผลของตัวเอง

> ด้วยความตรงไปตรงมาที่ผิดปกติ — "พูดตรงๆ ผมตื่นมาแล้วอารมณ์ไม่ดี เลยตัดสินใจว่าใครบางคนไม่ควรได้รับอนุญาตให้อยู่บนอินเทอร์เน็ต" เขาเขียน "ไม่มีใครควรมีอำนาจแบบนั้น" (Gizmodo, "Cloudflare CEO on Terminating Service to Neo-Nazi Site," 2017) ไม่มีศาลใดสั่งให้ถอดเว็บนี้ออก ไม่มีสภานิติบัญญัติใดลงมติเรื่องนี้ ผู้บริหารคนเดียวในบริษัทโครงสร้างพื้นฐานเพียงบริษัทเดียวกดสวิตช์ครั้งเดียว แล้วสิ่งพิมพ์ที่รอดพ้นความโกรธแค้นของสาธารณชนมาหลายปีก็หายไปจากอินเทอร์เน็ตที่เข้าถึงได้ภายในไม่กี่ชั่วโมง Prince พูดถูกที่ว่าไม่มีใครควรมีอำนาจแบบนั้น — และถูกด้วยที่ว่าในขณะนั้น เขาเองคือคนที่มีอำนาจนั้นอยู่จริง

Lawrence Lessig had already named the mechanism a decade earlier. *Code and Other Laws of Cyberspace* (1999, revised as *Code 2.0* in 2006) argues that four forces regulate behavior — law, social norms, the market, and architecture — and that in digital environments, architecture usually wins, because it does not ask for compliance, it makes the disfavored action impossible rather than illegal (Lessig, *Code and Other Laws of Cyberspace*). A law against a website can be appealed, delayed, ignored pending review. A DNS provider withdrawing service is not a rule about the website; it is the website's disappearance, instantly and everywhere, the moment the change propagates. Cloudflare's decision is Lessig's thesis in its purest form: code did not enforce the law here, code *was* the only law that mattered, and it was written by a company's terms of service rather than by any legislature.

> Lawrence Lessig ตั้งชื่อกลไกนี้ไว้แล้วสิบปีก่อนหน้า Code and Other Laws of Cyberspace (1999 ปรับปรุงเป็น Code 2.0 ในปี 2006) โต้แย้งว่ามีแรงสี่แรงที่กำกับพฤติกรรม — กฎหมาย บรรทัดฐานทางสังคม ตลาด และสถาปัตยกรรม (architecture) — และในสภาพแวดล้อมดิจิทัล สถาปัตยกรรมมักชนะเสมอ เพราะมันไม่ได้ขอให้ปฏิบัติตาม แต่ทำให้การกระทำที่ไม่พึงประสงค์เป็นไปไม่ได้เลยแทนที่จะแค่ผิดกฎหมาย (Lessig, Code and Other Laws of Cyberspace) กฎหมายที่ต่อต้านเว็บไซต์หนึ่งยังอุทธรณ์ได้ ยืดเวลาได้ เพิกเฉยระหว่างรอทบทวนได้ แต่ผู้ให้บริการ DNS ที่ถอนบริการไม่ใช่กฎเกี่ยวกับเว็บไซต์นั้น — มันคือการหายไปของเว็บไซต์นั้นเอง ทันทีและทุกที่ ในวินาทีที่การเปลี่ยนแปลงแพร่กระจายออกไป การตัดสินใจของ Cloudflare คือวิทยานิพนธ์ของ Lessig ในรูปแบบที่บริสุทธิ์ที่สุด — โค้ดไม่ได้

> บังคับใช้กฎหมายในกรณีนี้ โค้ด คือ กฎหมายเพียงฉบับเดียวที่มีความหมาย และมันถูกเขียนขึ้นโดยข้อกำหนดการให้บริการของบริษัทหนึ่ง ไม่ใช่โดยสภานิติบัญญัติใดเลย

Every major platform now runs something that functions like a legal system without being one. Kate Klonick's "The New Governors" describes Facebook, Twitter, and YouTube as private jurisdictions complete with legislation (terms of service), adjudication (content moderation review), and — since October 2020 — an appellate layer: Facebook's Oversight Board, a $130-million-endowed panel of twenty members whose rulings on content decisions are contractually binding on the company, and which overturned four of the first five decisions it reviewed (Klonick, "The New Governors," *Harvard Law Review* 131, 2018). It is jurisprudence's full apparatus, minus the one thing that ordinarily justifies it: no user elected the legislature, and the constitution can be rewritten unilaterally by the company that wrote it the first time. Prince's discomfort with his own power was, in effect, an admission that he had briefly become one of Klonick's "new governors" without any of the checks a real government answers to.

> ทุกวันนี้แพลตฟอร์มใหญ่ทุกเจ้าเดินระบบบางอย่างที่ทำหน้าที่เหมือนระบบกฎหมาย โดยไม่ได้เป็นระบบกฎหมายจริง บทความ "The New Governors" ของ Kate Klonick บรรยาย Facebook, Twitter และ YouTube ว่าเป็นเขตอำนาจศาลเอกชนที่ครบเครื่อง — มีนิติบัญญัติ (ข้อกำหนดการให้บริการ) มีการตัดสินคดี (การพิจารณาควบคุมเนื้อหา) และตั้งแต่เดือนตุลาคม 2020 มีชั้นอุทธรณ์ด้วย — Oversight Board ของ Facebook คณะกรรมการยี่สิบคนที่มีเงินทุนตั้งต้น 130 ล้านดอลลาร์ ซึ่งคำวินิจฉัยเรื่องเนื้อหาของคณะนี้ผูกพันบริษัทตามสัญญา และคณะนี้พลิกคำตัดสินสี่ในห้าคดีแรกที่พิจารณา (Klonick, "The New Governors," Harvard Law Review 131, 2018) มันคือกลไกนิติศาสตร์เต็มรูปแบบ ลบสิ่งเดียวที่ปกติทำให้กลไกแบบนี้ชอบธรรมออกไป — ไม่มีผู้ใช้คนไหนเลือกตั้งฝ่ายนิติบัญญัตินี้ และรัฐธรรมนูญของมันถูกเขียนใหม่ฝ่ายเดียวได้โดยบริษัทที่เขียนมันขึ้นมาตั้งแต่แรก ความอึดอัดของ Prince ต่ออำนาจของตัวเอง ที่จริงแล้วคือการยอมรับว่าเขาได้กลายเป็นหนึ่งใน "ผู้ปกครองใหม่ (new governors)" ของ Klonick อยู่ชั่วขณะหนึ่ง โดยไม่มีการตรวจสอบถ่วงดุลแบบที่รัฐบาลจริงต้องรับผิดชอบเลยแม้แต่น้อย

The deeper stake, Shoshana Zuboff argues, is not who moderates speech but who extracts and owns the "behavioral surplus" generated by everyone's use of these systems — the data left over once a service has been rendered, repurposed to predict and shape future behavior and sold as a prediction product to advertisers and beyond. *The Age of Surveillance Capitalism* (2019) contends this is a genuinely new form of power, distinct from the industrial capitalism that classical liberal theory built its protections against, because it operates through unilateral claims on experience that the subject never consented to and typically cannot see, let alone exit (Zuboff, *The Age of Surveillance Capitalism*, PublicAffairs, 2019). Contract law and privacy torts were built to police visible transactions between two parties who both know an exchange occurred; they were not built for a form of extraction whose whole design depends on the other party not noticing.

> Shoshana Zuboff โต้แย้งว่าเดิมพันที่ลึกกว่านั้นไม่ใช่เรื่องใครควบคุมคำพูด แต่คือใครสกัดและเป็นเจ้าของ "ส่วนเกินเชิงพฤติกรรม (behavioral surplus)" ที่เกิดจากการใช้งานระบบเหล่านี้ของทุกคน — ข้อมูลที่เหลือหลังบริการถูกส่งมอบแล้ว ถูกนำกลับมาใช้ใหม่เพื่อทำนายและกำหนดพฤติกรรมในอนาคต แล้วขายเป็นสินค้าทำนายให้กับผู้ลงโฆษณาและอื่นๆ ต่อไปอีก The Age of Surveillance Capitalism (2019) เสนอว่านี่คืออำนาจรูปแบบใหม่จริงๆ ที่แตกต่างจากทุนนิยมอุตสาหกรรมที่ทฤษฎีเสรีนิยมคลาสสิกสร้างการป้องกันขึ้นมาต่อต้าน เพราะมันทำงานผ่านการอ้างสิทธิ์ฝ่ายเดียวเหนือประสบการณ์ที่เจ้าของประสบการณ์ไม่เคยยินยอมและมักมองไม่เห็นด้วยซ้ำ นับประสาอะไรกับการถอนตัวออกมา (Zuboff, The Age of Surveillance Capitalism, PublicAffairs, 2019) กฎหมายสัญญาและการละเมิดเรื่องความเป็นส่วนตัวถูกสร้างขึ้นมาเพื่อกำกับดูแลธุรกรรมที่มองเห็นได้ระหว่างสองฝ่ายที่ต่างรู้ว่ามีการแลกเปลี่ยนเกิดขึ้น มันไม่ได้ถูกสร้างมาสำหรับการสกัดรูปแบบที่ทั้งการออกแบบขึ้นอยู่กับการที่อีกฝ่ายไม่รู้ตัว

Not every digital institution has defaulted to sovereign-or-corporate. Elinor Ostrom's *Governing the Commons* (1990) identified design principles by which real communities manage shared resources without either a state's coercion or a market's pricing — clearly bounded membership, rules matched to local conditions, graduated sanctions, cheap and accessible

conflict resolution, and governance nested across scales — overturning the assumption, inherited from Garrett Hardin, that shared resources are doomed to overuse absent private ownership or state control (Ostrom, *Governing the Commons*, 1990). Ostrom and Charlotte Hess later extended the same principles explicitly to knowledge and digital commons, arguing that open-source projects, Wikipedia, and shared research infrastructure could be read and governed the same way (Hess & Ostrom, eds., *Understanding Knowledge as a Commons*, MIT Press, 2007). Linux and Wikipedia are the proof this third path is not merely theoretical: neither a state nor a corporation, and yet governed well enough to outlast most of both.

> ไม่ใช่สถาบันดิจิทัลทุกแห่งที่ลงเอยแบบรัฐหรือบรรษัทเสมอไป Governing the Commons (1990) ของ Elinor Ostrom ระบุหลักการออกแบบที่ชุมชนจริงๆ ใช้จัดการทรัพยากรร่วม (commons) โดยไม่ต้องพึ่งทั้งการบังคับของรัฐและการตั้งราคาของตลาด — สมาชิกภาพที่มีขอบเขตชัดเจน กฎที่สอดคล้องกับสภาพท้องถิ่น บทลงโทษแบบขั้นบันได กระบวนการระงับข้อพิพาทที่ถูกและเข้าถึงง่าย และธรรมาภิบาลที่ซ้อนกันข้ามระดับ — ล้มล้างสมมติฐานที่สืบทอดมาจาก Garrett Hardin ที่ว่าทรัพยากรร่วมมีชะตากรรมต้องถูกใช้เกินขนาดหากไม่มีกรรมสิทธิ์เอกชนหรือการควบคุมของรัฐ (Ostrom, Governing the Commons, 1990) Ostrom และ Charlotte Hess ขยายหลักการเดียวกันนี้ในภายหลังไปสู่ความรู้และทรัพยากรร่วมดิจิทัลอย่างชัดเจน โดยโต้แย้งว่าโปรเจกต์โอเพนซอร์ส Wikipedia และโครงสร้างพื้นฐานการวิจัยที่ใช้ร่วมกันสามารถอ่านและปกครองได้ในแบบเดียวกัน (Hess & Ostrom, eds., Understanding Knowledge as a Commons, MIT Press, 2007) Linux และ Wikipedia คือหลักฐานว่าเส้นทางที่สามนี้ไม่ใช่แค่ทฤษฎี — ไม่ใช่ทั้งรัฐและไม่ใช่บรรษัท แต่ก็ยังถูกปกครองได้ดีพอจะอยู่ยั้งยืนยงกว่าทั้งสองแบบส่วนใหญ่

When a state itself delegates to code, the failure mode changes shape entirely. Danielle Citron's "Technological Due Process" (2008) warns that automated systems collapse two functions administrative law has always kept separate — writing the general rule and applying it to one person's case — into a single, undocumented act, evading the notice, hearing, and reasoned explanation either function alone would owe a citizen (Citron, "Technolo-

gical Due Process," *Washington University Law Review* 85, 2008). Michigan's unemployment agency ran exactly this experiment between 2013 and 2015: its MiDAS system flagged fraud automatically, ran start-to-finish without human review in most cases, and falsely accused more than 34,000 people, at an error rate later estimated at 85 percent, triggering wage garnishments and bankruptcies (IEEE Spectrum, "Michigan's MiDAS Unemployment System"). No rulemaking notice, no individual hearing, no appeal built in — due process was not violated by any one decision; it had been designed out of the system before the first case was ever flagged.

> เมื่อรัฐเองมอบหมายอำนาจให้โค้ด รูปแบบความล้มเหลวก็เปลี่ยนไปทั้งหมด บทความ "Technological Due Process" (2008) ของ Danielle Citron เตือนว่า ระบบอัตโนมัติยุบสองหน้าที่ที่กฎหมายปกครองแยกจากกันมาตลอด — การเขียนกฎทั่วไปกับการนำกฎไปใช้กับคดีของคนคนหนึ่ง — ให้เหลือเป็นการกระทำเดียวที่ไม่มีบันทึกไว้ หลบเลี่ยงการแจ้งเตือน การไต่สวน และคำอธิบายที่มีเหตุผลซึ่งแต่ละหน้าที่เพียงลำพังพึงต้องมอบให้พลเมือง (Citron, "Technological Due Process," Washington University Law Review 85, 2008) หน่วยงานประกันการว่างงานของรัฐมิชิแกนรันการทดลองแบบนี้เป๊ะๆ ระหว่างปี 2013 ถึง 2015 — ระบบ MiDAS ของหน่วยงานนี้ตั้งธงการฉ้อโกงโดยอัตโนมัติ รันตั้งแต่ต้นจนจบโดยไม่มีมนุษย์ทบทวนในกรณีส่วนใหญ่ และกล่าวหาผิดคนไปกว่า 34,000 คน ด้วยอัตราความผิดพลาดที่ประเมินภายหลังได้ที่ 85 percent จนนำไปสู่การอายัดค่าจ้างและการล้มละลาย (IEEE Spectrum, "Michigan's MiDAS Unemployment System") ไม่มีการแจ้งเตือนการออกกฎ ไม่มีการไต่สวนรายบุคคล ไม่มีการอุทธรณ์ในตัวระบบ — กระบวนการที่ชอบด้วยกฎหมาย (due process) ไม่ได้ถูกละเมิดโดยการตัดสินใจครั้งใดครั้งหนึ่ง — มันถูกออกแบบให้ขาดหายไปจากระบบตั้งแต่ก่อนคดีแรกจะถูกตั้งธงด้วยซ้ำ

Benjamin Bratton's *The Stack* (2015) offers the frame that holds Cloudflare, Facebook, Zuboff's harvesters, and Michigan's benefits algorithm as one phenomenon rather than four unrelated scandals: planetary computation — cloud, city, address, interface, user, layered as a single accidental megastructure — now performs sovereign functions (deciding who may speak, who receives a public benefit, who is reachable at all) continuously

and at a scale no state apparatus can match, without ever standing for election or answering to a constitution (Bratton, *The Stack*, MIT Press, 2015). The state has not been abolished by the stack. It has been joined by a second, unelected one, and most days it is not obvious which one actually decided.

> The Stack (2015) ของ Benjamin Bratton เสนอกรอบที่รวม Cloudflare, Facebook, เครื่องเก็บเกี่ยวข้อมูลของ Zuboff และอัลกอริทึมสวัสดิการของมิชิแกนไว้เป็นปรากฏการณ์เดียวกัน แทนที่จะเป็นเรื่องอื้อฉาวสี่เรื่องที่ไม่เกี่ยวข้องกัน — การคำนวณระดับดาวเคราะห์ (planetary computation) — คลาวด์ เมือง ที่อยู่อินเทอร์เฟซ ผู้ใช้ ซ้อนกันเป็นมหาโครงสร้างที่เกิดขึ้นโดยบังเอิญเพียงหนึ่งเดียว — ตอนนี้ทำหน้าที่แบบอธิปไตย (ตัดสินว่าใครพูดได้ ใครได้รับสวัสดิการสาธารณะ ใครเข้าถึงได้เลยหรือไม่) อย่างต่อเนื่องและในสเกลที่กลไกรัฐใดก็เทียบไม่ได้ โดยไม่เคยต้องลงสมัครรับเลือกตั้งหรือรับผิดชอบต่อรัฐธรรมนูญใดเลย (Bratton, The Stack, MIT Press, 2015) รัฐไม่ได้ถูกยกเลิกไปโดยสแตกนี้ มันแค่มีสิ่งที่สองที่ไม่ได้มาจากการเลือกตั้งมาสมทบด้วย และเกือบทุกวันก็ไม่ชัดเจนด้วยซ้ำว่าฝ่ายไหนกันแน่ที่เป็นผู้ตัดสินใจจริง

Connections · เชื่อมโยง

Chapter 06 owns the responsibility gap this chapter's automated systems keep opening, asking who is accountable rather than who governs. Chapter 11 explains the bracket that lets "behavioral surplus" be priced as a quantity while what it is *of* goes unasked. Chapter 14 picks up Ostrom's commons again through consensus protocols — cooperation among strangers with no sovereign to enforce it.

> บทที่ 06 เป็นเจ้าของช่องว่างความรับผิดชอบที่ระบบอัตโนมัติในบทนี้เปิดขึ้นซ้ำแล้วซ้ำเล่า โดยถามว่าใครรับผิดชอบ ไม่ใช่ใครปกครอง บทที่ 11 อธิบายวงเล็บที่ทำให้ "ส่วนเกินเชิงพฤติกรรม (behavioral surplus)" ถูกตั้งราคาเป็นปริมาณได้ ในขณะที่ไม่มีใครถามว่ามันคือส่วนเกิน ของ อะไรกันแน่ บทที่ 14 หยิบทรัพยากรร่วม (commons) ของ Ostrom กลับมาอีกครั้งผ่านโพรโทคอลฉันทามติ (consensus protocol) — ความร่วมมือระหว่างคนแปลกหน้าโดยไม่มีผู้มีอธิปไตยคอยบังคับใช้

Open questions · คำถามที่ยังเปิดอยู่

Is "code is law" a warning about what has already happened, or a program for what should happen next — and does confusing the two let more power slide into architecture than anyone actually voted for? Can a private platform hold due-process-like authority legitimately, short of becoming a state in miniature complete with the checks that make states bearable? And does Ostrom's commons model generalize to physical infrastructure — cables, satellites, spectrum — the way it generalized from irrigation to code, or does infrastructure's need for capital always pull governance back toward a state or a firm?

> "โค้ดคือกฎหมาย (code is law)" เป็นคำเตือนเรื่องสิ่งที่เกิดขึ้นไปแล้ว หรือเป็นแผนงานสำหรับสิ่งที่ควรเกิดขึ้นต่อไป — และการสับสนระหว่างสองอย่างนี้ทำให้อำนาจเลื่อนไหลเข้าไปอยู่ในสถาปัตยกรรมมากกว่าที่ใครเคยลงคะแนนให้จริงหรือไม่ แพลตฟอร์มเอกชนถืออำนาจแบบกระบวนการที่ชอบด้วยกฎหมายได้อย่างชอบธรรมหรือไม่ โดยไม่ต้องกลายเป็นรัฐจำลองที่มีกลไกตรวจสอบถ่วงดุลครบแบบที่ทำให้รัฐพอทนได้ และโมเดลทรัพยากรร่วมของ Ostrom ขยายไปใช้กับโครงสร้างพื้นฐานทางกายภาพ — สายเคเบิล ดาวเทียม คลื่นความถี่ — ได้แบบเดียวกับที่มันขยายจากการชลประทานไปสู่โค้ดหรือไม่ หรือความต้องการเงินทุนของโครงสร้างพื้นฐานจะดึงธรรมาภิบาลกลับไปหารัฐหรือบริษัทเสมอ

Go deeper · อ่านต่อ

— Lawrence Lessig, *Code: Version 2.0*, Basic Books, 2006 — the revised edition of the argument this chapter's opening case runs on rails built by.

> Lawrence Lessig, Code: Version 2.0, Basic Books, 2006 — ฉบับปรับปรุงของข้อโต้แย้งที่กรณีเปิดเรื่องของบทนี้วิ่งอยู่บนรางที่มันวางไว้

— Kate Klonick, "The New Governors: The People, Rules, and Processes Governing Online Speech," *Harvard Law Review* 131 (2018), read against Elinor Ostrom & Charlotte Hess, eds., *Understanding Knowledge as a Commons: From Theory to Practice*, MIT Press, 2007 — private jurisprudence against the commons it displaced.

> Kate Klonick, "The New Governors: The People, Rules, and Processes Governing Online Speech," Harvard Law Review 131 (2018), read against Elinor Ostrom & Charlotte Hess, eds., Understanding Knowledge as a Commons: From Theory to Practice, MIT Press, 2007 — นิติศาสตร์เอกชนเทียบกับทรัพยากรร่วมที่มันเข้ามาแทนที่

— Stanford Encyclopedia of Philosophy, "Political Legitimacy" — the standing account of what a governing power owes the governed, against which every case in this chapter can be measured.

> Stanford Encyclopedia of Philosophy, "Political Legitimacy" — บันทึกหลักเรื่องสิ่งที่อำนาจปกครองพึงมีต่อผู้ถูกปกครอง ซึ่งทุกกรณีในบทนี้วัดเทียบกับมันได้

CHAPTER 08

Aesthetics & Phenomenology

สุนทรียศาสตร์และปรากฏการณ์วิทยา

This chapter sits in Part I, the Court of Philosophy — the eighth and final hearing before the benches switch, where philosophy asks the machine what the critics and the artists notice that the builders do not.

บทนี้อยู่ในภาค I ศาลปรัชญา — การไต่สวนครั้งที่ 8 อันเป็นครั้งสุดท้ายก่อนม้านั่งคู่ความจะสลับฝั่ง ที่ปรัชญาตั้งคำถามกับเครื่องจักรว่านักวิจารณ์และศิลปินสังเกตเห็นอะไรที่ผู้สร้างมองไม่เห็น

What do the critics and the artists see that the builders miss?

นักวิจารณ์และศิลปินเห็นอะไรที่ผู้สร้างมองข้ามไป

In October 2018, Christie's auctioned *Portrait of Edmond de Belamy*, a blurred, unfinished-looking image of a stout man in a dark coat, estimated at $7,000–10,000. It sold for $432,500. The work had been produced by a generative adversarial network trained by Obvious, a three-person Paris collective, on roughly 15,000 portraits scraped from WikiArt — and, as reporters soon discovered, on code adapted almost unmodified from a public GitHub repository written by Robbie Barrat, a teenager Obvious never publicly credited until pressed (Hyperallergic, "Christie's Sells 'AI-Generated' Art for $432,500 As Controversy Swirls Over Creators' Use of Copied Code"). Where a painter signs a canvas, Obvious printed, in the lower-right corner, the mathematical objective their network had trained against: $\min_G \max_D \mathbb{E}_x[\log(D(x))] + \mathbb{E}_z[\log(1-D(G(z)))]$ — the generator and discriminator's loss function, standing in for a hand. The painting had, quite literally, signed itself with its own machinery instead of an author.

> ในเดือนตุลาคม 2018 Christie's ประมูล Portrait of Edmond de Belamy ภาพเบลอๆ ดูเหมือนวาดไม่เสร็จของชายร่างท้วมในเสื้อโค้ตสีเข้ม ตั้งราคาประเมินไว้ที่ 7,000–10,000 ดอลลาร์ แต่ขายไปได้ที่ 432,500 ดอลลาร์ ผลงานชิ้นนี้ผลิตขึ้นโดย

> generative adversarial network ที่ Obvious กลุ่มศิลปินสามคนจากปารีสเทรนขึ้นจากภาพเหมือนราว 15,000 ภาพที่ดึงมาจาก WikiArt — และตามที่นักข่าวค้นพบในเวลาต่อมา ยังใช้โค้ดที่ดัดแปลงมาแทบไม่เปลี่ยนแปลงจากที่เก็บ GitHub สาธารณะที่เขียนโดย Robbie Barrat วัยรุ่นคนหนึ่งที่ Obvious ไม่เคยให้เครดิตต่อสาธารณะจนกว่าจะถูกกดดัน (Hyperallergic, "Christie's Sells 'AI-Generated' Art for $432,500 As Controversy Swirls Over Creators' Use of Copied Code") ในจุดที่จิตรกรจะเซ็นชื่อบนผืนผ้าใบ Obvious พิมพ์เป้าหมายทางคณิตศาสตร์ที่เครือข่ายของพวกเขาเทรนเทียบไว้ที่มุมล่างขวาแทน — min G max D 𝔼x[log(D(x))] + 𝔼z[log(1−D(G(z)))] — ฟังก์ชันการสูญเสีย (loss function) ของ generator กับ discriminator ทำหน้าที่แทนลายมือ ภาพวาดนี้เซ็นชื่อตัวเองด้วยกลไกของมันเองแทนที่จะเป็นผู้ประพันธ์ ตามตัวอักษรจริงๆ

Every question this chapter has to ask is already sitting in that signature. Whose work is it — Obvious's, for choosing the dataset and the frame and the price? Barrat's, for writing the network that actually learned the weights? The eighteenth- and nineteenth-century portraitists in WikiArt, whose brushwork the model had statistically absorbed and recombined? Or no one's, because a loss function has no hand at all? The builders of computing have their own, older answer to a related question — not who made a thing beautiful, but why beauty should matter to engineering at all — and it is worth hearing their case before the critics take the stand.

> ทุกคำถามที่บทนี้ต้องถามนั่งอยู่ในลายเซ็นนั้นแล้ว ผลงานนี้เป็นของใคร — ของ Obvious เพราะเลือกชุดข้อมูล กรอบ และราคา ของ Barrat เพราะเป็นคนเขียนเครือข่ายที่เรียนรู้ค่าน้ำหนัก (weight) จริงๆ ของจิตรกรผู้วาดภาพเหมือนในศตวรรษที่สิบแปดและสิบเก้าใน WikiArt ที่ฝีแปรงของพวกเขาถูกโมเดลดูดซับและผสมผสานใหม่ทางสถิติ หรือไม่เป็นของใครเลย เพราะฟังก์ชันการสูญเสียไม่มีมือเลยแม้แต่น้อย ผู้สร้างวงการคอมพิวเตอร์มีคำตอบของตัวเอง ที่เก่ากว่า ต่อคำถามที่เกี่ยวข้องกัน — ไม่ใช่ใครทำให้สิ่งหนึ่งงดงาม แต่ทำไมความงามถึงควรมีความหมายต่อวิศวกรรมเลย — และควรฟังข้อโต้แย้งฝ่ายนี้ก่อนที่นักวิจารณ์จะขึ้นให้การ

G. H. Hardy opened A *Mathematician's Apology* (1940) with a claim that sounds like vanity and is actually an epistemic demand: "Beauty is the first test: there is no permanent place in the world for ugly mathematics" (Hardy, A *Mathematician's Apology*, 1940). Computing inherited the demand and gave it teeth. Edsger Dijkstra's 1968 letter to *Communications of the ACM*, "Go To Statement Considered Harmful," argued that unrestrained jumps in a program's control flow make it impossible for a reader to hold a correctness argument in mind — not merely ugly, but unverifiable, because a mind that cannot trace a program's structure cannot check its proof. Donald Knuth, who had begun the project in 1962, published the first volume of *The Art of Computer Programming* that same year and has spent more than six decades still writing it, volume by volume, because for Knuth an algorithm's elegance and its correctness were never two separate virtues to trade against each other. In this tradition, beauty is not decoration applied after the logic works. It is the condition under which a human being — not just a compiler — can tell that the logic works at all.

G. H. Hardy เปิด A Mathematician's Apology (1940) ด้วยข้อกล่าวอ้างที่ฟังดูเหมือนความหลงตัวเอง แต่จริงๆ แล้วคือข้อเรียกร้องเชิงญาณวิทยา (epistemic) — "ความงามคือบททดสอบข้อแรก โลกนี้ไม่มีที่ทางถาวรให้กับคณิตศาสตร์ที่น่าเกลียด" ("Beauty is the first test: there is no permanent place in the world for ugly mathematics") (Hardy, A Mathematician's Apology, 1940) วงการคอมพิวเตอร์รับข้อเรียกร้องนี้มาสืบทอดแล้วใส่เขี้ยวเล็บให้มัน จดหมายปี 1968 ของ Edsger Dijkstra ถึง Communications of the ACM เรื่อง "Go To Statement Considered Harmful" โต้แย้งว่าการกระโดดข้ามที่ไม่มีข้อจำกัดใน control flow ของโปรแกรมทำให้ผู้อ่านถือข้อโต้แย้งเรื่องความถูกต้องไว้ในหัวไม่ได้ — ไม่ใช่แค่น่าเกลียด แต่ตรวจสอบไม่ได้เลย เพราะจิตที่ไล่ตามโครงสร้างของโปรแกรมไม่ได้ ก็ตรวจบทพิสูจน์ของมันไม่ได้เช่นกัน Donald Knuth ผู้เริ่มโปรเจกต์นี้ตั้งแต่ปี 1962 ตีพิมพ์เล่มแรกของ The Art of Computer Programming ในปีเดียวกันนั้น และใช้เวลากว่าหกทศวรรษเขียนมันต่อไปเรื่อยๆ ทีละเล่ม เพราะสำหรับ Knuth ความสง่างามของอัลกอริทึมกับความถูกต้องของมันไม่เคยเป็นคุณธรรมสองอย่างที่ต้องแลกกันเลย ใน

> ประเพณีนี้ ความงามไม่ใช่เครื่องประดับที่แปะเข้าไปหลังตรรกะทำงานได้แล้ว มันคือเงื่อนไขที่ทำให้มนุษย์คนหนึ่ง — ไม่ใช่แค่คอมไพเลอร์ — บอกได้ว่าตรรกะนั้นทำงานได้จริงหรือเปล่า

Hubert Dreyfus thought the builders were solving the wrong problem entirely. In a 1965 RAND paper commissioned to evaluate artificial intelligence research and titled, pointedly, "Alchemy and Artificial Intelligence," he argued that treating cognition as rule-governed symbol manipulation missed what Heidegger and Merleau-Ponty had already shown about ordinary human competence: skillful coping in a world — knowing how to finish a sentence, cross a room, read a face — is holistic, embodied, and pre-theoretical, not decomposable into the explicit rules a program could follow (RAND, *Alchemy and Artificial Intelligence*, P-3244, 1965). The expanded argument, *What Computers Can't Do* (1972), was met with outrage from the AI community it targeted and quietly vindicated by a discipline that had not yet been founded. Rodney Brooks's subsumption robots, described in "Elephants Don't Play Chess" (1990), abandoned symbolic world-models entirely for layered, reactive behaviors coupled directly to sensors and motors — Brooks's own slogan, "the world is its own best model," is close enough to Dreyfus's phenomenology that embodied robotics reads, in hindsight, like Dreyfus's critique built rather than merely argued. Large language models complicate the picture from the opposite direction: fluent, contextually apt conversation produced by systems with no body, no world to cope in, and nothing at stake in any outcome — an achievement Dreyfus's argument, on its face, should not have permitted. Whether that is a refutation or a change of subject remains open: fluent text is not obviously the same accomplishment as skillful worldly coping, and defenders of the critique — including Dreyfus's own later essay "Why Heideggerian AI Failed" (2007) — would say the goalposts moved rather than the argument broke.

> Hubert Dreyfus คิดว่าผู้สร้างกำลังแก้ปัญหาผิดข้อทั้งหมด ในเปเปอร์ RAND ปี 1965 ที่ได้รับมอบหมายให้ประเมินงานวิจัยปัญญาประดิษฐ์ และตั้งชื่อไว้อย่างจงใจว่า "Alchemy and Artificial Intelligence" เขาโต้แย้งว่าการมองการรู้คิด (cognition) เป็นการจัดการสัญลักษณ์ที่ถูกกำกับด้วยกฎ พลาดสิ่งที่ Heidegger และ

> Merleau-Ponty แสดงให้เห็นไปแล้วเกี่ยวกับความสามารถของมนุษย์ธรรมดา — การรับมือกับโลกอย่างชำนาญ ไม่ว่าจะเป็นการรู้วิธีพูดประโยคให้จบ เดินข้ามห้อง หรืออ่านสีหน้าคน — เป็นเรื่ององค์รวม มีร่าง (embodied) และก่อนทฤษฎี ไม่สามารถแยกย่อยลงเป็นกฎชัดแจ้งที่โปรแกรมจะทำตามได้ (RAND, Alchemy and Artificial Intelligence, P-3244, 1965) ข้อโต้แย้งฉบับขยาย What Computers Can't Do (1972) เจอกับความโกรธเกรี้ยวจากวงการ AI ที่มันพุ่งเป้าเล่นงาน และได้รับการพิสูจน์ว่าถูกต้องอย่างเงียบๆ จากสาขาวิชาที่ยังไม่ถือกำเนิดขึ้นในตอนนั้น หุ่นยนต์แบบ subsumption ของ Rodney Brooks ที่บรรยายไว้ใน "Elephants Don't Play Chess" (1990) ทิ้งแบบจำลองโลกเชิงสัญลักษณ์ไปทั้งหมด หันไปใช้พฤติกรรมแบบเป็นชั้นๆ ที่ตอบสนองทันทีและเชื่อมตรงกับเซนเซอร์และมอเตอร์แทน — คำขวัญของ Brooks เองที่ว่า "โลกคือแบบจำลองที่ดีที่สุดของตัวมันเอง" ใกล้เคียงกับปรากฏการณ์วิทยาของ Dreyfus มากพอที่หุ่นยนต์แบบมีร่าง (embodied robotics) เมื่อมองย้อนกลับไปแล้ว จะอ่านได้เหมือนข้อวิจารณ์ของ Dreyfus ถูกสร้างขึ้นจริง ไม่ใช่แค่ถูกโต้แย้งไว้ โมเดลภาษาขนาดใหญ่ทำให้ภาพซับซ้อนขึ้นจากทิศตรงข้าม — บทสนทนาที่คล่องแคล่วและเข้ากับบริบท ผลิตโดยระบบที่ไม่มีร่างกาย ไม่มีโลกให้ต้องรับมือ และไม่มีอะไรเป็นเดิมพันในผลลัพธ์ใดเลย — เป็นความสำเร็จที่ข้อโต้แย้งของ Dreyfus เมื่อมองผิวเผินแล้วไม่ควรจะอนุญาตให้เกิดขึ้นได้ นี่เป็นการหักล้างหรือแค่เปลี่ยนหัวข้อยังคงเป็นคำถามเปิด — ข้อความที่คล่องแคล่วไม่ใช่ความสำเร็จเดียวกันอย่างชัดเจนกับการรับมือโลกอย่างชำนาญ และผู้ปกป้องข้อวิจารณ์นี้ — รวมถึงบทความยุคหลังของ Dreyfus เองเรื่อง "Why Heideggerian AI Failed" (2007) — จะบอกว่าเป้าหมายถูกย้าย ไม่ใช่ข้อโต้แย้งพัง

Heidegger's own later work supplies the vocabulary for what happened to Barrat's code once it entered Obvious's pipeline. "The Question Concerning Technology" (1954, developed from a 1949 Bremen lecture) argues that modern technology's essence is a mode of revealing he calls *Gestell*, en-framing: the world shows up not as something encountered on its own terms but as *standing-reserve* (*Bestand*) — resource, on tap, awaiting further ordering (Wikipedia, "Gestell"). Barrat's repository, released in the spirit of open-source sharing, was absorbed by Obvious as exactly this: not an authored work to be encountered and credited, but raw material on tap for a

saleable print. This is also the moment this chapter owes its reader the fact usually left out of Heidegger's aesthetics: he joined the Nazi Party on 1 May 1933, ten days after being elected Rector of Freiburg, delivered a rectoral address endorsing the regime, and never fully accounted for either afterward. A philosopher who diagnosed the ordering of the world into mere resource is not thereby cleared of having served, at the one moment his institutional power was greatest, a regime that ordered human beings into exactly that category.

> งานยุคหลังของ Heidegger เองเป็นผู้ให้คำศัพท์สำหรับสิ่งที่เกิดขึ้นกับโค้ดของ Barrat เมื่อมันเข้าสู่กระบวนการผลิตของ Obvious "The Question Concerning Technology" (1954 พัฒนาจากการบรรยายที่ Bremen ในปี 1949) โต้แย้งว่าแก่นแท้ของเทคโนโลยีสมัยใหม่คือรูปแบบหนึ่งของการเปิดเผยที่เขาเรียกว่า Gestell การจัดกรอบ (enframing) — โลกไม่ได้ปรากฏตัวในฐานะสิ่งที่ถูกพบเจอตามเงื่อนไขของมันเอง แต่เป็น คลังสำรองพร้อมใช้ (standing-reserve) (Bestand) — ทรัพยากรที่พร้อมจ่าย รอการจัดระเบียบต่อไป (Wikipedia, "Gestell") ที่เก็บโค้ดของ Barrat ที่เผยแพร่ด้วยจิตวิญญาณของการแบ่งปันแบบโอเพนซอร์ส ถูก Obvious ดูดซับไปเป็นแบบนี้เป๊ะๆ — ไม่ใช่ผลงานที่มีผู้ประพันธ์ซึ่งควรถูกพบเจอและให้เครดิต แต่เป็นวัตถุดิบดิบที่พร้อมจ่ายสำหรับงานพิมพ์ที่ขายได้ นี่ยังเป็นจุดที่บทนี้ติดค้างข้อเท็จจริงกับผู้อ่านที่มักถูกละไว้จากสุนทรียศาสตร์ของ Heidegger — เขาเข้าร่วมพรรคนาซีในวันที่ 1 พฤษภาคม 1933 สิบวันหลังได้รับเลือกเป็นอธิการบดีของ Freiburg กล่าวสุนทรพจน์รับตำแหน่งอธิการบดีที่สนับสนุนระบอบนั้น และไม่เคยอธิบายทั้งสองเรื่องนี้ได้อย่างครบถ้วนในภายหลัง นักปรัชญาที่วินิจฉัยการจัดระเบียบโลกให้กลายเป็นแค่ทรัพยากรนั้น ไม่ได้พ้นผิดจากการรับใช้ระบอบที่จัดระเบียบมนุษย์ให้กลายเป็นหมวดหมู่แบบเดียวกันเป๊ะ ในช่วงเวลาที่อำนาจในสถาบันของเขาสูงสุด

Andy Clark and David Chalmers's "The Extended Mind" (1998) offers, unexpectedly, the frame that ties the signature back to the hammer. Their parity principle holds that a part of the world counts as part of a cognitive process if it plays the role a part of the brain would play were the process run entirely in the head — illustrated by Otto, their thought-experiment Alzheimer's patient, whose notebook functions as his memory precisely because he consults it as unreflectively as a person consults recollection

(Clark & Chalmers, "The Extended Mind," *Analysis* 58(1), 1998). That transparency-in-use is close to Heidegger's earlier and more famous idea, from *Being and Time* (1927): a hammer, skillfully wielded, withdraws from attention and becomes *ready-to-hand* (*zuhanden*); only a broken hammer becomes *present-at-hand* (*vorhanden*), a mere object demanding inspection. Elegant code disappears into a reader's understanding the same way; Otto's notebook disappears into his remembering the same way; and a programmer pairing with a language model that completes half their keystrokes is asking, in practice, exactly Otto's question — where does the boundary of authorship sit once part of the work happens somewhere the hand never touched. Obvious signed their canvas with an equation because, whether they knew it or not, that was the one honest answer available: the hand had already withdrawn.

> "The Extended Mind" (1998) ของ Andy Clark และ David Chalmers ให้กรอบที่ไม่คาดคิดมาก่อน ซึ่งผูกลายเซ็นนั้นกลับไปหาค้อนได้ หลักความเสมอภาค (parity principle) ของพวกเขายืนยันว่าส่วนหนึ่งของโลกนับเป็นส่วนหนึ่งของกระบวนการรู้คิดได้ ถ้ามันเล่นบทบาทที่ส่วนหนึ่งของสมองจะเล่นหากกระบวนการนั้นทำงานอยู่ในหัวทั้งหมด — แสดงให้เห็นผ่าน Otto ผู้ป่วยอัลไซเมอร์ในการทดลองทางความคิดของพวกเขา ที่สมุดบันทึกทำหน้าที่เป็นความทรงจำของเขาเป๊ะๆ เพราะเขาปรึกษามันโดยไม่ต้องคิดทบทวน เหมือนที่คนคนหนึ่งปรึกษาความทรงจำของตัวเอง (Clark & Chalmers, "The Extended Mind," Analysis 58(1), 1998) ความโปร่งใสในการใช้งานแบบนั้นใกล้เคียงกับไอเดียยุคก่อนหน้าและมีชื่อเสียงกว่าของ Heidegger จาก Being and Time (1927) — ค้อนที่ถูกใช้งานอย่างชำนาญจะถอนตัวออกจากความสนใจและกลายเป็น พร้อมมือ (ready-to-hand) (zuhanden) มีแต่ค้อนที่หักเท่านั้นที่กลายเป็น ประจักษ์ต่อหน้า (present-at-hand) (vorhanden) วัตถุธรรมดาที่เรียกร้องการตรวจสอบ โค้ดที่สง่างามหายไปในความเข้าใจของผู้อ่านแบบเดียวกัน สมุดบันทึกของ Otto หายไปในการจดจำของเขาแบบเดียวกัน และโปรแกรมเมอร์ที่ทำงานคู่กับโมเดลภาษาที่เติมคีย์สโตรกให้ครึ่งหนึ่งกำลังถามคำถามเดียวกับของ Otto ในทางปฏิบัติ — เส้นแบ่งของความเป็นผู้ประพันธ์อยู่ตรงไหนกันแน่ เมื่อส่วน

> หนึ่งของงานเกิดขึ้นในที่ที่มือไม่เคยแตะเลย Obvious เซ็นชื่อบนผืนผ้าใบของพวกเขาด้วยสมการ เพราะไม่ว่าพวกเขาจะรู้ตัวหรือไม่ นั่นคือคำตอบที่ซื่อสัตย์เพียงคำตอบเดียวที่มีอยู่ — มือได้ถอนตัวออกไปแล้ว

Connections · เชื่อมโยง

Chapter 04 supplies the deeper argument about symbols, minds, and the Chinese Room that Dreyfus's critique runs alongside. Chapter 05 returns to Barrat's code through Wittgenstein's rule-following problem — what a repository "says" once someone else runs it. Chapter 13 continues the embodiment question through the grounding problem in learning machines.

> บทที่ 04 ให้ข้อโต้แย้งที่ลึกกว่านั้นเกี่ยวกับสัญลักษณ์ จิต และห้องจีน (Chinese Room) ที่ข้อวิจารณ์ของ Dreyfus วิ่งขนานไปด้วย บทที่ 05 กลับไปหาโค้ดของ Barrat ผ่านปัญหาการทำตามกฎ (rule-following) ของ Wittgenstein — ที่เก็บหนึ่ง "พูด" อะไรเมื่อมีคนอื่นรันมัน บทที่ 13 สานต่อคำถามเรื่องการมีร่าง (embodiment) ผ่านปัญหาการยึดโยง (grounding) ในเครื่องจักรเรียนรู้

Open questions · คำถามที่ยังเปิดอยู่

If elegance is a precondition for verifiability rather than decoration, does an unreadable but provably correct proof — the kind machines increasingly produce — count as beautiful in Hardy's sense, or has that criterion quietly stopped applying? Was Dreyfus's critique refuted by fluent language models, or did fluency simply stop being the test that mattered? And once authorship can be distributed across a dataset, a borrowed repository, a curatorial choice, and a signature that is itself an equation, is "who made this" still a question with one right answer?

> ถ้าความสง่างามเป็นเงื่อนไขล่วงหน้าของการตรวจสอบได้ (verifiability) ไม่ใช่เครื่องประดับ บทพิสูจน์ที่อ่านไม่ออกแต่พิสูจน์ได้ว่าถูกต้อง — แบบที่เครื่องจักรผลิตออกมามากขึ้นเรื่อยๆ — นับเป็นความงามในความหมายของ Hardy หรือไม่ หรือเกณฑ์นั้นเลิกใช้ได้อย่างเงียบๆ ไปแล้ว ข้อวิจารณ์ของ Dreyfus ถูกหักล้างโดยโมเดลภาษาที่พูดคล่องแคล่วหรือไม่ หรือความคล่องแคล่วแค่เลิกเป็นบททดสอบที่สำคัญไป

เฉยๆ และเมื่อความเป็นผู้ประพันธ์กระจายออกไปได้ทั้งชุดข้อมูล ที่เก็บที่หยิบยืมมา ทางเลือกเชิงภัณฑารักษ์ (curatorial choice) และลายเซ็นที่ตัวมันเองก็คือสมการ "ใครทำสิ่งนี้" ยังเป็นคำถามที่มีคำตอบถูกต้องเพียงหนึ่งเดียวอยู่หรือไม่

Go deeper · อ่านต่อ

- Donald Knuth, *The Art of Computer Programming*, Addison-Wesley, begun 1962, first volume published 1968 — the canonical case that correctness and elegance were never two different virtues.

  > Donald Knuth, The Art of Computer Programming, Addison-Wesley, begun 1962, first volume published 1968 — กรณีตัวอย่างหลักที่ว่าความถูกต้องกับความสง่างามไม่เคยเป็นคุณธรรมสองอย่างที่แยกจากกัน

- Hubert L. Dreyfus, "Alchemy and Artificial Intelligence," RAND Corporation, P-3244, 1965, read against Andy Clark & David J. Chalmers, "The Extended Mind," *Analysis* 58(1), 1998 — the critique of disembodied cognition against the theory that lets a notebook, or a network, count as part of a mind.

  > Hubert L. Dreyfus, "Alchemy and Artificial Intelligence," RAND Corporation, P-3244, 1965, read against Andy Clark & David J. Chalmers, "The Extended Mind," Analysis 58(1), 1998 — ข้อวิจารณ์การรู้คิดแบบไร้ร่างกาย เทียบกับทฤษฎีที่ยอมให้สมุดบันทึกหรือเครือข่ายหนึ่งนับเป็นส่วนหนึ่งของจิตได้

- Stanford Encyclopedia of Philosophy, "Martin Heidegger" — the standing account of enframing, readiness-to-hand, and the biography neither can be read apart from.

  > Stanford Encyclopedia of Philosophy, "Martin Heidegger" — บันทึกหลักเรื่องการจัดกรอบ ความพร้อมมือ และประวัติชีวิตที่ทั้งสองเรื่องแยกอ่านออกจากกันไม่ได้

CHAPTER 09

Computability

การคำนวณได้

This is the first hearing of Part II. Computation stops being the witness and takes the examiner's chair — and its first question is aimed at itself: what does its own founding limit actually prove?

นี่คือการไต่สวนครั้งแรกของภาค II การคำนวณเลิกเป็นพยานและขึ้นนั่งเก้าอี้ผู้ซักถามแทน — และคำถามแรกของมันพุ่งเข้าใส่ตัวมันเอง: ขีดจำกัดที่เป็นรากฐานของมันเองพิสูจน์อะไรได้จริง?

What do the limits of computation prove about the limits of reason?

ขีดจำกัดของการคำนวณพิสูจน์อะไรเกี่ยวกับขีดจำกัดของเหตุผล?

Two papers settled the same question six weeks apart in 1936, and neither author had read the other's proof. Alonzo Church's "An Unsolvable Problem of Elementary Number Theory" appeared on 15 April in the *American Journal of Mathematics* (58: 345–363), showing by lambda-definability that no algorithm decides the *Entscheidungsproblem* — the question, posed by David Hilbert and Wilhelm Ackermann in *Grundzüge der theoretischen Logik* (1928), of whether a mechanical procedure could certify the validity of any first-order sentence. Alan Turing's own paper, "On Computable Numbers, with an Application to the Entscheidungsproblem," was received by the *Proceedings of the London Mathematical Society* on 28 May — and reached the identical negative answer by an entirely different route: a human clerk idealized down to a paper tape, a finite set of states, and a table of instructions. Turing, disappointed to learn he had been beaten to publication, added an appendix proving that his machines and Church's lambda-definable functions compute exactly the same class of functions. Two definitions of

"mechanical," invented independently, turned out to coincide exactly. That coincidence is the whole of the evidence for what is now called the Church–Turing thesis, and evidence is not the same thing as proof.

> สองงานวิจัยตอบคำถามเดียวกันห่างกันแค่หกสัปดาห์ในปี 1936 โดยที่ผู้เขียนแต่ละคนไม่เคยอ่านบทพิสูจน์ของอีกฝ่าย บทความ "An Unsolvable Problem of Elementary Number Theory" ของ Alonzo Church ตีพิมพ์วันที่ 15 เมษายน ใน American Journal of Mathematics (58: 345–363) แสดงผ่าน lambda-definability ว่าไม่มีอัลกอริทึมใดตัดสิน Entscheidungsproblem (ปัญหาการตัดสินใจ) ได้ — คำถามที่ David Hilbert และ Wilhelm Ackermann ตั้งไว้ใน Grundzüge der theoretischen Logik (1928) ว่ามีกระบวนการเชิงกลไกที่รับรองความสมเหตุสมผลของประโยคอันดับหนึ่ง (first-order sentence) ใดๆ ได้หรือไม่ บทความของ Alan Turing เอง "On Computable Numbers, with an Application to the Entscheidungsproblem" ถูกรับโดย Proceedings of the London Mathematical Society วันที่ 28 พฤษภาคม — และไปถึงคำตอบเชิงลบแบบเดียวกันด้วยเส้นทางที่ต่างออกไปโดยสิ้นเชิง: เสมียนมนุษย์ที่ถูกอุดมคติลงเหลือเพียงเทปกระดาษ ชุดสถานะจำกัดชุดหนึ่ง และตารางคำสั่ง Turing รู้สึกผิดหวังที่รู้ว่าตัวเองถูกตีพิมพ์แซงหน้า จึงเพิ่มภาคผนวกพิสูจน์ว่าเครื่องของเขากับฟังก์ชัน lambda-definable ของ Church คำนวณฟังก์ชันคลาสเดียวกันเป๊ะ นิยามของ "เชิงกลไก" สองแบบ ที่คิดขึ้นแยกกันโดยอิสระ กลับตรงกันพอดิบพอดี ความบังเอิญนั้นคือหลักฐานทั้งหมดที่มีสำหรับสิ่งที่ปัจจุบันเรียกว่าข้อตั้ง Church–Turing (Church–Turing thesis) และหลักฐานไม่ใช่สิ่งเดียวกันกับบทพิสูจน์

It cannot be a theorem, because "effectively calculable" was never a formal term before Turing formalized it — you cannot prove an identity between something precise and something that, until that moment, was only an intuition. What Turing actually did was analyze, by exhaustion, everything a human "computer" — a person following fixed instructions with pencil and paper — could do: read a symbol, change an internal state, write a symbol, shift attention. Every rival formalism proposed since — Gödel–Herbrand's general recursive functions, Emil Post's production systems, register machines — has been proven equivalent to Turing's, and Stephen Kleene, who

named the conjecture "Church's Thesis" in 1943, treated that convergence as inductive support rather than deductive closure. So the thesis sits in a category Hilbert's program never anticipated: not quite mathematics, not quite empirical science, held up by the fact that everyone who has tried to break it has instead confirmed it.

> มันเป็นทฤษฎีบทไม่ได้ เพราะ "คำนวณได้อย่างมีประสิทธิผล" (effectively calculable) ไม่เคยเป็นคำนิยามเชิงรูปนัยมาก่อนที่ Turing จะทำให้มันเป็นรูปนัย — จะพิสูจน์ความเท่ากันระหว่างสิ่งที่แม่นยำกับสิ่งที่จนถึงตอนนั้นยังเป็นแค่สัญชาตญาณไม่ได้ สิ่งที่ Turing ทำจริงๆ คือวิเคราะห์แบบไล่ให้ครบทุกกรณีว่ามนุษย์ที่เป็น "computer" — คนที่ทำตามคำสั่งตายตัวด้วยดินสอกับกระดาษ — ทำอะไรได้บ้าง: อ่านสัญลักษณ์ เปลี่ยนสถานะภายใน เขียนสัญลักษณ์ ย้ายความสนใจ ทุกฟอร์มัลลิสม์ (formalism) คู่แข่งที่เสนอมาตั้งแต่นั้น — ฟังก์ชันเวียนบังเกิดทั่วไป (general recursive functions) ของ Gödel–Herbrand, ระบบการผลิต (production systems) ของ Emil Post, เครื่องรีจิสเตอร์ (register machines) — ล้วนถูกพิสูจน์แล้วว่าเทียบเท่ากับของ Turing และ Stephen Kleene ผู้ตั้งชื่อข้อคาดการณ์นี้ว่า "Church's Thesis" ในปี 1943 มองว่าการลู่เข้าหากันนั้นเป็นแค่หลักฐานสนับสนุนเชิงอุปนัย ไม่ใช่การปิดกั้นเชิงนิรนัย ข้อตั้งนี้จึงอยู่ในหมวดหมู่ที่โครงการของ Hilbert ไม่เคยคาดคิดมาก่อน: ไม่ใช่คณิตศาสตร์เต็มตัว ไม่ใช่วิทยาศาสตร์เชิงประจักษ์เต็มตัว ยืนอยู่ได้ด้วยข้อเท็จจริงที่ว่าทุกคนที่พยายามหักล้างมันกลับยืนยันมันแทน

That instability is itself philosophically productive, because the method Turing used to reach it — diagonalization, turning a system's own machinery against itself — outgrew the theorem it built. Cantor used it in 1891 to show no list of real numbers can be complete; Gödel adapted it in 1931 to build an arithmetical sentence that asserts its own unprovability; Turing adapted it again to show no algorithm can decide, for an arbitrary program, whether it halts — feed a hypothetical halting-decider its own description and a contradiction falls out in one line. The same skeleton reappears in Tarski's proof that no sufficiently rich language can define its own truth predicate, and in Frederic Fitch's 1963 paradox of knowability. Diagonalization stopped being a mathematician's trick around the middle of the twenti-

eth century and became philosophy's standard proof that a system cannot fully turn its lights on itself — the argument form this book's own courtroom spine depends on.

> ความไม่มั่นคงนั้นเองกลับให้ผลผลิตเชิงปรัชญา เพราะวิธีที่ Turing ใช้ไปถึงมัน — การทแยงมุม (diagonalization) คือการหันกลไกของระบบมาโจมตีตัวมันเอง — เติบโตเกินกว่าทฤษฎีบทที่มันสร้างขึ้นมาเสียอีก Cantor ใช้มันในปี 1891 เพื่อแสดงว่าไม่มีรายการจำนวนจริงใดครบถ้วนได้ Gödel ดัดแปลงมันในปี 1931 เพื่อสร้างประโยคเลขคณิตที่ยืนยันความพิสูจน์ไม่ได้ของตัวมันเอง Turing ดัดแปลงมันอีกครั้งเพื่อแสดงว่าไม่มีอัลกอริทึมใดตัดสินปัญหาการหยุด (halting problem) ได้ สำหรับโปรแกรมใดก็ตามจะบอกไม่ได้ว่ามันจะหยุดทำงานหรือไม่ — ป้อนตัวตัดสิน halting problem สมมติด้วยคำอธิบายของตัวมันเอง แล้วความขัดแย้งก็โผล่ออกมาในบรรทัดเดียว โครงเดียวกันนี้ปรากฏซ้ำในบทพิสูจน์ของ Tarski ที่ว่าไม่มีภาษาใดสมบูรณ์พอจะนิยามภาคแสดงความจริง (truth predicate) ของตัวมันเองได้ และในปฏิทรรศน์ความรู้ได้ (paradox of knowability) ของ Frederic Fitch ปี 1963 diagonalization เลิกเป็นแค่กลเม็ดของนักคณิตศาสตร์ราวกลางศตวรรษที่ 20 แล้วกลายเป็นบทพิสูจน์มาตรฐานของปรัชญาที่ว่าระบบหนึ่งเปิดไฟส่องดูตัวเองให้ทั่วทั้งหมดไม่ได้ — รูปแบบข้อโต้แย้งที่แกนกลางห้องพิจารณาคดีของหนังสือเล่มนี้เองพึ่งพาอยู่

It is also the argument form behind the most publicized claim in this territory. J. R. Lucas told the Oxford Philosophical Society in 1959, and then argued in print in "Minds, Machines and Gödel" (*Philosophy* 36, 1961: 112–127), that Gödel's theorem refutes mechanism: for any consistent formal system F strong enough to encode arithmetic, Gödel's construction yields a true sentence F cannot prove, yet a human reasoner, standing outside F and reflecting on it, can see that the sentence is true — so no single formal system captures everything a mind can know, and minds are therefore not machines. Roger Penrose revived the argument at greater length in *The Emperor's New Mind* (1989) and defended a reinforced version in *Shadows of the Mind* (1994). The standard reply, developed across decades of responses from Paul Benacerraf onward, is that the argument smuggles in exactly the certainty Gödel's *second* incompleteness theorem forbids: to "see" that the

Gödel sentence for F is true, the reasoner must already know F is consistent, and no consistent system — human or mechanical — can prove its own consistency from within. The argument also assumes a human mind is well modeled as one single, fixed, consistent formal system in the first place, which nothing here establishes. Computation, examining this claim on its own witness stand, cannot certify it; but philosophy answers back here too, because the machine's own founding thesis has the identical shape of unprovable self-trust — the Church–Turing thesis cannot certify its own domain without appealing to an intuition about "mechanical" that no formalism proves, any more than the human reasoner can certify F's consistency from inside F.

> นี่ก็เป็นรูปแบบข้อโต้แย้งเดียวกันที่อยู่เบื้องหลังข้อกล่าวอ้างที่โด่งดังที่สุดในอาณาเขตนี้ J. R. Lucas พูดต่อหน้า Oxford Philosophical Society ในปี 1959 แล้วเขียนโต้แย้งเป็นลายลักษณ์อักษรใน "Minds, Machines and Gödel" (Philosophy 36, 1961: 112–127) ว่าทฤษฎีบทของ Gödel หักล้างกลไกนิยม (mechanism): สำหรับระบบรูปนัย F ที่สมเหตุสมผลใดๆ ที่แข็งแรงพอจะเข้ารหัสเลขคณิตได้ การสร้างของ Gödel ให้ประโยคจริงประโยคหนึ่งที่ F พิสูจน์ไม่ได้ แต่ผู้ให้เหตุผลที่เป็นมนุษย์ ซึ่งยืนอยู่นอก F และไตร่ตรองมัน กลับเห็นได้ว่าประโยคนั้นจริง — ฉะนั้นไม่มีระบบรูปนัยระบบเดียวใดจับทุกสิ่งที่จิตรู้ได้ทั้งหมด และจิตจึงไม่ใช่เครื่องจักร Roger Penrose รื้อฟื้นข้อโต้แย้งนี้ให้ยาวขึ้นใน The Emperor's New Mind (1989) และปกป้องเวอร์ชันที่แข็งแรงขึ้นใน Shadows of the Mind (1994) คำตอบโต้มาตรฐานที่พัฒนาผ่านหลายทศวรรษของการตอบโต้นับตั้งแต่ Paul Benacerraf เป็นต้นมา คือข้อโต้แย้งนี้แอบสอดใส่ความแน่นอนแบบเดียวกับที่ทฤษฎีบทความไม่สมบูรณ์ข้อที่สองของ Gödel ห้ามไว้พอดี: การจะ "เห็น" ว่าประโยค Gödel สำหรับ F เป็นจริง ผู้ให้เหตุผลต้องรู้อยู่ก่อนแล้วว่า F สมเหตุสมผล และไม่มีระบบที่สมเหตุสมผล — ไม่ว่ามนุษย์หรือเครื่องจักร — พิสูจน์ความสมเหตุสมผลของตัวมันเองจากภายในได้ ข้อโต้แย้งนี้ยังสมมติด้วยว่าจิตมนุษย์ถูกจำลองได้ดีในฐานะระบบรูปนัยหนึ่งเดียวตายตัว สมเหตุสมผล ตั้งแต่แรก ซึ่งไม่มีอะไรในที่นี้ยืนยันเรื่องนั้น การคำนวณ เมื่อตรวจสอบข้อกล่าวอ้างนี้จากที่นั่งพยานของตัวมันเอง รับรองมันไม่ได้ แต่ปรัชญาก็ตอบโต้กลับตรงนี้เช่นกัน เพราะข้อตั้งก่อตั้งของเครื่องจักรเองก็มีรูปทรงเดียวกันกับ

> ความไว้ใจตัวเองที่พิสูจน์ไม่ได้ — Church–Turing thesis รับรองอาณาเขตของตัวมันเองไม่ได้โดยไม่อาศัยสัญชาตญาณเกี่ยวกับ "เชิงกลไก" ที่ไม่มีฟอร์มัลลิสม์ใดพิสูจน์ได้ ไม่ต่างจากที่ผู้ให้เหตุผลมนุษย์รับรองความสมเหตุสมผลของ F จากภายใน F ไม่ได้

Turing knew his own thesis had an edge, because he drew it himself. His 1939 Princeton doctoral thesis, "Systems of Logic Based on Ordinals," introduced oracle machines: Turing machines equipped with a black-box oracle answering questions no Turing machine could resolve unaided. B. Jack Copeland and Diane Proudfoot later named the research program built on that opening "hypercomputation," and Copeland has argued that exotic physical setups — closed timelike curves, or the infinite-proper-time computations permitted in a Malament–Hogarth spacetime — might in principle realize an oracle. The honest objection is the verification problem: even if such a device were built, nothing from outside could ever confirm it had computed a genuinely non-Turing-computable function rather than an extremely large Turing-computable one. Here physics, not logic, casts the deciding vote on whether the thesis bounds all possible computation or only computation as our universe happens to permit it.

> Turing เองก็รู้ว่าข้อตั้งของตัวเองมีขอบอยู่ เพราะเขาเป็นคนขีดมันเอง วิทยานิพนธ์ปริญญาเอกที่ Princeton ปี 1939 ของเขา "Systems of Logic Based on Ordinals" แนะนำเครื่องออราเคิล (oracle machine): เครื่องทัวริงที่ติดตั้งออราเคิลกล่องดำที่ตอบคำถามซึ่งเครื่องทัวริงเครื่องใดก็ตอบเองไม่ได้ B. Jack Copeland และ Diane Proudfoot ตั้งชื่อโครงการวิจัยที่สร้างขึ้นจากจุดเริ่มนั้นในภายหลังว่าไฮเปอร์คอมพิวเตชัน (hypercomputation) และ Copeland โต้แย้งว่าโครงสร้างทางฟิสิกส์ที่แปลกประหลาด — closed timelike curve หรือการคำนวณแบบ infinite-proper-time ที่กาลอวกาศแบบ Malament–Hogarth เปิดทางให้ — อาจตระหนักออราเคิลได้จริงในหลักการ ข้อคัดค้านที่ตรงไปตรงมาคือปัญหาการตรวจสอบยืนยัน (verification problem): ต่อให้สร้างอุปกรณ์แบบนั้นขึ้นมาได้จริง ก็ไม่มีอะไรจากภายนอกยืนยันได้เลยว่ามันคำนวณฟังก์ชันที่ non-Turing-computable แท้ๆ ไม่ใช่ฟังก์ชันที่ Turing-computable แต่ใหญ่โตมโหฬารเท่านั้น ตรงนี้ฟิสิกส์ ไม่ใช่ตรรกศาสตร์ เป็นผู้ลงคะแนนตัดสินว่าข้อตั้งนี้ครอบคลุมการคำนวณที่เป็นไปได้ทั้งหมด หรือครอบคลุมแค่การคำนวณเท่าที่เอกภพของเราเผอิญอนุญาตให้ทำได้

Gödel himself licensed far less than his popularizers claim. In his 1951 Gibbs Lecture he stated a disjunction and stopped there: either the human mind is not equivalent to any finite machine, or there exist mathematical problems absolutely undecidable by any means — or both — but he refused to say which, remaining a mathematical Platonist who suspected the first disjunct without ever asserting he had proved it. What Gödel proved is a limit inside formal systems; what Lucas, Penrose, and a century of secondary literature have drawn from it is a conclusion about minds that the theorem itself never reaches.

> Gödel เองอนุญาตไว้น้อยกว่าที่คนเผยแพร่ผลงานของเขาอ้างมาก ในการบรรยาย Gibbs Lecture ปี 1951 เขาระบุเงื่อนไขแบบ "หรือ" (disjunction) ไว้แล้วหยุดอยู่ตรงนั้น: ไม่ว่าจิตมนุษย์จะไม่เทียบเท่ากับเครื่องจักรจำกัดใดๆ หรือมีปัญหาทางคณิตศาสตร์ที่ตัดสินไม่ได้เด็ดขาดไม่ว่าด้วยวิธีใด — หรือทั้งสองอย่าง — แต่เขาปฏิเสธที่จะบอกว่าข้อไหน ยังคงเป็นนักเพลโตนิยมทางคณิตศาสตร์ที่สงสัยตัวเลือกแรกโดยไม่เคยยืนยันว่าตนพิสูจน์มันได้แล้ว สิ่งที่ Gödel พิสูจน์คือขีดจำกัดภายในระบบรูปนัย ส่วนสิ่งที่ Lucas, Penrose และวรรณกรรมรองตลอดหนึ่งศตวรรษดึงออกมาจากมันคือข้อสรุปเกี่ยวกับจิตที่ตัวทฤษฎีบทเองไม่เคยไปถึง

Connections · เชื่อมโยง

Chapter 00 named 1936 as the moment the machine fell out of a philosophical proof; this chapter returns to ask what that moment actually licenses. Chapter 01 owns Gödel and Curry–Howard as logic's shared inheritance. Chapter 04 owns the question the Lucas–Penrose argument tries to settle from outside — whether the mind is what the brain computes. Chapter 10 tightens "in principle" into a measurable resource. Chapter 12 shows what proof assistants do with incompleteness once it is treated as an engineering constraint rather than a philosophical verdict.

> บทที่ 00 เรียกปี 1936 ว่าคือช่วงที่เครื่องจักรหล่นออกมาจากบทพิสูจน์เชิงปรัชญา บทนี้ย้อนกลับไปถามว่าช่วงเวลานั้นอนุญาตอะไรไว้จริงๆ บทที่ 01 เป็นเจ้าของ Gödel และ Curry–Howard ในฐานะมรดกร่วมของตรรกศาสตร์ บทที่ 04 เป็นเจ้าของคำถามที่ข้อโต้แย้งของ Lucas–Penrose พยายามตัดสินจากภายนอก — ว่า

> จิตคือสิ่งที่สมองคำนวณหรือไม่ บทที่ 10 บีบ "ในหลักการ" ให้กลายเป็นทรัพยากรที่วัดได้ บทที่ 12 แสดงให้เห็นว่า proof assistant ทำอะไรกับความไม่สมบูรณ์ เมื่อมันถูกมองเป็นข้อจำกัดเชิงวิศวกรรมแทนที่จะเป็นคำตัดสินเชิงปรัชญา

Open questions · คำถามที่ยังเปิดอยู่

Is the Church–Turing thesis falsifiable — and would a working hypercomputer refute it, or merely relocate the boundary it draws? Does Gödel's 1951 disjunction ever get resolved, or does it stay open the way the thesis itself does? And if minds turn out to be Turing-equivalent, does that settle functionalism, or only computability — a question this chapter can raise but not answer.

> Church–Turing thesis พิสูจน์เท็จได้ (falsifiable) หรือไม่ — และถ้ามี hypercomputer ที่ใช้งานได้จริง มันจะหักล้างข้อตั้งนี้ หรือแค่ย้ายเส้นขอบที่มันขีดไว้เท่านั้น? เงื่อนไขแบบ "หรือ" ของ Gödel ในปี 1951 เคยถูกคลี่คลายไหม หรือยังคงเปิดค้างเหมือนที่ตัวข้อตั้งเองยังเปิดค้างอยู่? และถ้าจิตกลายเป็นสิ่งที่ Turing-equivalent จริง นั่นจะยุติข้อถกเถียงเรื่องหน้าที่นิยม (functionalism) หรือยุติแค่เรื่องการคำนวณได้เท่านั้น — คำถามที่บทนี้ตั้งได้แต่ตอบไม่ได้

Go deeper · อ่านต่อ

— B. Jack Copeland, Carl J. Posy, and Oron Shagrir, eds., *Computability: Turing, Gödel, Church, and Beyond* (Cambridge, MA: MIT Press, 2013).

> B. Jack Copeland, Carl J. Posy, and Oron Shagrir, eds., Computability: Turing, Gödel, Church, and Beyond (Cambridge, MA: MIT Press, 2013).

— Alonzo Church, "An Unsolvable Problem of Elementary Number Theory," *American Journal of Mathematics* 58 (1936): 345–363; Alan Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, ser. 2, 42 (1936–37): 230–265, received 28 May 1936 — doi.org/10.1112/plms/s2-42.1.230.

> Alonzo Church, "An Unsolvable Problem of Elementary Number Theory," American Journal of Mathematics 58 (1936): 345–363; Alan Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," Proceedings of the London Mathematical Society, ser. 2, 42 (1936–37): 230–265, received 28 May 1936 — doi.org/10.1112/plms/s2-42.1.230.

— Stanford Encyclopedia of Philosophy, "The Church-Turing Thesis" — plato.stanford.edu/entries/church-turing.

> Stanford Encyclopedia of Philosophy, "The Church-Turing Thesis" — plato.stanford.edu/entries/church-turing.

CHAPTER 10

Complexity

ความซับซ้อน

Second hearing of Part II. Chapter 09 asked what computation can do at all; this one asks what it can do before the universe runs out of patience — and whether that second question is merely practical or cuts as deep as any distinction philosophy has drawn.

การไต่สวนครั้งที่สองของภาค II บทที่ 09 ถามว่าการคำนวณทำอะไรได้บ้างในหลักการ บทนี้ถามว่ามันทำอะไรได้ก่อนที่เอกภพจะหมดความอดทน — และคำถามที่สองนี้เป็นแค่เรื่องเชิงปฏิบัติ หรือลึกพอๆ กับข้อแบ่งแยกใดๆ ที่ปรัชญาเคยขีดไว้

Is "possible in principle but not in practice" a metaphysical category?

"เป็นไปได้ในหลักการแต่ไม่เป็นไปได้ในทางปฏิบัติ" เป็นหมวดหมู่เชิงอภิปรัชญาหรือไม่?

In May 1971, at the third ACM Symposium on Theory of Computing, Stephen Cook presented "The Complexity of Theorem-Proving Procedures," proving that Boolean satisfiability — given a formula, does some assignment of true/false to its variables make it true? — has a peculiar property: every problem whose proposed solutions can be *checked* quickly reduces to it. Two years later, cut off from Western journals, Leonid Levin proved the identical theorem in the Soviet Union without having seen Cook's paper — a second independent convergence on the same result, the same shape as Church and Turing thirty-five years earlier. Richard Karp's 1972 paper "Reducibility Among Combinatorial Problems" then showed that twenty-one familiar hard problems — the travelling salesman, graph coloring, clique — are all satisfiability in disguise. Cook, Levin, and Karp had located a single fault line: P, the class of problems solvable quickly, against NP, the class whose

solutions can be checked quickly once found. Whether P equals NP is the most consequential open question in mathematics, and almost every expert believes the answer is no.

> เดือนพฤษภาคม 1971 ในงาน ACM Symposium on Theory of Computing ครั้งที่สาม Stephen Cook นำเสนอ "The Complexity of Theorem-Proving Procedures" พิสูจน์ว่า Boolean satisfiability — กำหนดสูตรตรรกะมาหนึ่งสูตร มีการกำหนดค่าจริง/เท็จให้ตัวแปรของมันแบบใดแบบหนึ่งที่ทำให้สูตรนั้นเป็นจริงหรือไม่? — มีคุณสมบัติประหลาดอย่างหนึ่ง: ทุกปัญหาที่คำตอบที่เสนอมาสามารถตรวจสอบได้อย่างรวดเร็ว ล้วนลดรูปมาเป็นปัญหานี้ได้ทั้งหมด สองปีต่อมา Leonid Levin ซึ่งถูกตัดขาดจากวารสารตะวันตก พิสูจน์ทฤษฎีบทเดียวกันเป๊ะในสหภาพโซเวียตโดยไม่เคยเห็นบทความของ Cook มาก่อน — การลู่เข้าหากันโดยอิสระครั้งที่สองสู่ผลลัพธ์เดียวกัน รูปทรงเดียวกับ Church และ Turing เมื่อสามสิบห้าปีก่อนหน้านั้น บทความของ Richard Karp ปี 1972 เรื่อง "Reducibility Among Combinatorial Problems" จากนั้นแสดงว่าปัญหายากที่คุ้นเคย 21 ปัญหา — พนักงานขายเดินทาง (travelling salesman), การระบายสีกราฟ, คลีค (clique) — ล้วนเป็น satisfiability ที่แค่แปลงร่างมา Cook, Levin และ Karp ระบุรอยแยกเดียวได้: P คือคลาสของปัญหาที่แก้ได้อย่างรวดเร็ว เทียบกับ NP คือคลาสที่คำตอบของมันตรวจสอบได้อย่างรวดเร็วเมื่อหาเจอแล้ว P เท่ากับ NP หรือไม่คือคำถามเปิดที่มีผลสะเทือนมากที่สุดในคณิตศาสตร์ และผู้เชี่ยวชาญแทบทุกคนเชื่อว่าคำตอบคือไม่เท่ากัน

Scott Aaronson's 2011 manifesto "Why Philosophers Should Care About Computational Complexity" argues that this is not a technicality philosophy can set aside as an engineering detail. Classical epistemic logic assumes an agent who believes all the logical consequences of what she believes — logical omniscience — and treats the assumption as an idealization, the way physics idealizes frictionless planes. Complexity theory shows it is not an idealization but an impossibility: for many ordinary beliefs, deriving every consequence would outlast the universe even granted every computer humanity could ever build. Aaronson extends the same lens to Hume's prob-

lem of induction, to Newcomb's problem, and to economic rationality — in each case, a puzzle that looked purely conceptual turns out to have a computational skeleton nobody had priced in.

> แถลงการณ์ปี 2011 ของ Scott Aaronson เรื่อง "Why Philosophers Should Care About Computational Complexity" โต้แย้งว่านี่ไม่ใช่รายละเอียดทางเทคนิคที่ปรัชญาจะโละทิ้งไปเป็นเรื่องวิศวกรรมได้ ตรรกศาสตร์เชิงญาณ (epistemic logic) แบบคลาสสิกสมมติว่ามีผู้กระทำที่เชื่อในผลสืบเนื่องเชิงตรรกะทั้งหมดของสิ่งที่ตนเชื่อ — ความรอบรู้เชิงตรรกะ (logical omniscience) — และมองข้อสมมตินี้เป็นอุดมคติ แบบเดียวกับที่ฟิสิกส์อุดมคติระนาบไร้แรงเสียดทาน ทฤษฎีความซับซ้อนแสดงว่ามันไม่ใช่อุดมคติแต่เป็นไปไม่ได้เลย: สำหรับความเชื่อธรรมดาจำนวนมาก การอนุมานผลสืบเนื่องทุกข้อจะใช้เวลานานเกินอายุเอกภพ ต่อให้ยกคอมพิวเตอร์ทุกเครื่องที่มนุษยชาติเคยสร้างได้มาให้ทั้งหมดก็ตาม Aaronson ขยายมุมมองเดียวกันนี้ไปถึงปัญหาการอุปนัยของ Hume ปัญหาของ Newcomb และเหตุผลเชิงเศรษฐศาสตร์ — ในแต่ละกรณี ปริศนาที่ดูเหมือนเป็นแค่เรื่องเชิงมโนทัศน์ล้วนๆ กลับมีโครงเชิงคำนวณซ่อนอยู่ข้างในที่ไม่มีใครเคยตีราคาไว้ก่อน

The clearest case of that skeleton is what P vs NP says about creativity. Checking a completed Sudoku, or a proposed mathematical proof, takes a fixed, modest amount of work regardless of how the answer was found; searching the space of all possible arrangements or all possible proofs is — if P ≠ NP — exponentially harder in the worst case, no matter how cleverly the search is organized. A mathematician's flash of insight, the moment a proof strategy presents itself, may be doing something no systematic algorithm can replace: recognizing is cheap, finding is not, and the distance between them is precisely what complexity theory measures. Chess supplies an older, purer instance of the same gap. Ernst Zermelo proved in 1913 that chess is *determined* — one of "White forces a win," "Black forces a win," or "both force at worst a draw" is simply true, since the game tree is finite. Nobody knows which, because "finite" and "searchable" are different claims;

the game tree is too large for exhaustive analysis by any device anyone expects to build. "Solved in principle" and "true" turn out to be much further apart than either word suggests.

> กรณีที่ชัดที่สุดของโครงนี้คือสิ่งที่ P vs NP บอกเกี่ยวกับความคิดสร้างสรรค์ การตรวจสอบซูโดกุที่เติมเสร็จแล้ว หรือบทพิสูจน์ทางคณิตศาสตร์ที่เสนอมา ใช้แรงงานคงที่จำนวนไม่มากไม่ว่าคำตอบนั้นถูกหามาได้อย่างไร แต่การค้นหาในปริภูมิของการจัดเรียงที่เป็นไปได้ทั้งหมด หรือบทพิสูจน์ที่เป็นไปได้ทั้งหมด — ถ้า P ≠ NP — จะยากขึ้นแบบยกกำลัง (exponentially) ในกรณีเลวร้ายที่สุด ไม่ว่าจะจัดระเบียบการค้นหาให้ฉลาดแค่ไหนก็ตาม ประกายความเข้าใจของนักคณิตศาสตร์ ช่วงเวลาที่กลยุทธ์การพิสูจน์ผุดขึ้นมาเอง อาจกำลังทำสิ่งที่ไม่มีอัลกอริทึมเชิงระบบใดแทนที่ได้: การจำได้ว่าถูกนั้นถูกกว่า การหามันเจอนั้นไม่ถูกกว่า และระยะห่างระหว่างสองสิ่งนี้คือสิ่งที่ทฤษฎีความซับซ้อนวัดพอดี หมากรุกให้ตัวอย่างที่เก่าแก่กว่าและบริสุทธิ์กว่าของช่องว่างเดียวกันนี้ Ernst Zermelo พิสูจน์ในปี 1913 ว่าหมากรุกถูกกำหนดผลไว้แล้ว (determined) — หนึ่งในสามข้อ "ขาวบังคับให้ชนะได้" "ดำบังคับให้ชนะได้" หรือ "ทั้งสองฝ่ายบังคับได้อย่างน้อยที่สุดคือเสมอ" เป็นจริงอย่างแน่นอน เพราะต้นไม้เกม (game tree) ของมันจำกัด ไม่มีใครรู้ว่าข้อไหน เพราะ "จำกัด" กับ "ค้นหาได้ทั่วถึง" เป็นข้อกล่าวอ้างคนละเรื่องกัน ต้นไม้เกมใหญ่เกินกว่าจะวิเคราะห์ให้ครบทุกกรณีได้ด้วยอุปกรณ์ใดๆ ที่ใครจะคาดหวังให้สร้างขึ้นมาได้ "แก้ได้ในหลักการ" กับ "จริง" กลับห่างกันมากกว่าที่คำทั้งสองบอกเป็นนัยไว้มาก

Herbert Simon anticipated the gap from the other direction, decades before complexity theory could measure it. His *Administrative Behavior* (1947) and "A Behavioral Model of Rational Choice" (*Quarterly Journal of Economics* 69, 1955: 99–118) replaced the classical agent — who surveys every option and every consequence before choosing the best — with a *satisficer*, who stops at the first option clearing a "good enough" threshold. Simon meant this as psychology, a description of how managers and shoppers actually decide; complexity theory later gave it a floor that is not psychological at all. Any

agent operating under a real time budget — human, corporation, or algorithm — is a bounded-rational agent as a matter of physical necessity, whether or not it wants to be.

> Herbert Simon คาดการณ์ช่องว่างนี้ไว้จากอีกทิศทางหนึ่ง หลายทศวรรษก่อนที่ทฤษฎีความซับซ้อนจะวัดมันได้ หนังสือ Administrative Behavior (1947) และบทความ "A Behavioral Model of Rational Choice" (Quarterly Journal of Economics 69, 1955: 99–118) ของเขาแทนที่ผู้กระทำแบบคลาสสิก — ผู้สำรวจตัวเลือกและผลลัพธ์ทุกข้อก่อนเลือกตัวที่ดีที่สุด — ด้วยผู้กระทำแบบ satisficing (การพอใจแค่พอ) ผู้หยุดที่ตัวเลือกแรกที่ผ่านเกณฑ์ "ดีพอ" Simon ตั้งใจให้นี่เป็นจิตวิทยา คือคำอธิบายว่าผู้จัดการและคนซื้อของตัดสินใจกันจริงๆ อย่างไร ทฤษฎีความซับซ้อนในเวลาต่อมาให้พื้นให้มันซึ่งไม่ใช่เรื่องจิตวิทยาเลย ผู้กระทำใดก็ตามที่ทำงานภายใต้งบเวลาจริง — ไม่ว่าจะเป็นมนุษย์ บริษัท หรืออัลกอริทึม — ล้วนเป็นผู้กระทำที่มีเหตุผลมีขอบเขต (bounded rationality) โดยความจำเป็นทางกายภาพ ไม่ว่าจะต้องการเป็นแบบนั้นหรือไม่ก็ตาม

The same fault line, treated as a resource rather than an obstacle, built modern cryptography. Whitfield Diffie and Martin Hellman's "New Directions in Cryptography" (*IEEE Transactions on Information Theory* 22, 1976: 644–654) proposed that a "trapdoor one-way function" — easy to compute forward, computationally infeasible to invert without a secret key — could let two strangers agree on a shared secret over an open channel, though the paper offered no working example. RSA supplied one soon after: multiplying two three-hundred-digit primes takes a processor milliseconds; factoring the product back into its primes is believed, by every known method, to take longer than the universe has existed. Every padlock icon in a browser rests not on a proof but on an unproven complexity conjecture — one-way functions are, quite literally, the applied physics of keeping a secret.

> รอยแยกเดียวกันนี้ เมื่อถูกมองเป็นทรัพยากรแทนที่จะเป็นอุปสรรค กลับสร้างวิทยาการเข้ารหัสสมัยใหม่ขึ้นมา บทความ "New Directions in Cryptography" ของ Whitfield Diffie และ Martin Hellman (IEEE Transactions on Information

> Theory 22, 1976: 644–654) เสนอว่า ฟังก์ชันประตูกลทางเดียว (trapdoor one-way function) — คำนวณไปข้างหน้าง่าย แต่ผกผันกลับโดยไม่มีกุญแจลับแทบเป็นไปไม่ได้ในเชิงคำนวณ — น่าจะทำให้คนแปลกหน้าสองคนตกลงความลับร่วมกันได้ผ่านช่องทางเปิด แม้ตัวบทความจะไม่มีตัวอย่างที่ใช้งานได้จริงเสนอไว้ก็ตาม ไม่นานหลังจากนั้น RSA ก็ให้ตัวอย่างมา: การคูณจำนวนเฉพาะสามร้อยหลักสองตัวใช้เวลาโปรเซสเซอร์แค่มิลลิวินาที ส่วนการแยกตัวประกอบผลคูณกลับเป็นจำนวนเฉพาะเดิม เชื่อกันว่าด้วยวิธีที่รู้จักทุกวิธี จะใช้เวลานานกว่าอายุของเอกภพเสียอีก ไอคอนแม่กุญแจทุกอันบนเบราว์เซอร์วางอยู่บนข้อคาดการณ์เชิงความซับซ้อนที่ยังพิสูจน์ไม่ได้ ไม่ใช่บทพิสูจน์ — ฟังก์ชันทางเดียว (one-way function) คือฟิสิกส์ประยุกต์ของการเก็บความลับ ตามความหมายตรงตัวเป๊ะๆ

Complexity theory's strangest offspring reworks what a proof even is. Shafi Goldwasser, Silvio Micali, and Charles Rackoff's "The Knowledge Complexity of Interactive Proof-Systems" (17th ACM Symposium on Theory of Computing, Providence, 1985; full version, *SIAM Journal on Computing* 18, 1989: 186–208) showed that a prover can convince a verifier a statement is true — "these two graphs are isomorphic," "I know the password" — through an interactive back-and-forth whose transcript, read afterward by anyone who wasn't part of the exchange, proves nothing at all. The evidential force lives entirely in the live interrogation and its shrinking probability of letting a cheat through, not in a static object anyone can inspect later. That breaks with every proof from Euclid onward, where a proof is a fixed thing you hand someone and they check alone; here philosophy has grounds to object that computation has redefined "proof" to mean something merely statistical, and complexity theory's honest answer is that the older picture of certainty was never available at scale in the first place — chapter 12 finishes that argument.

> ลูกที่แปลกประหลาดที่สุดของทฤษฎีความซับซ้อนคือมันเขียนนิยามของคำว่า "บทพิสูจน์" ใหม่ทั้งหมด บทความ "The Knowledge Complexity of Interactive Proof-Systems" ของ Shafi Goldwasser, Silvio Micali และ Charles Rackoff (17th ACM Symposium on Theory of Computing, Providence, 1985; ฉบับเต็มใน SIAM Journal on Computing 18, 1989: 186–208) แสดงว่าผู้พิสูจน์ (prover) สามารถ

> โน้มน้าวผู้ตรวจสอบ (verifier) ให้เชื่อว่าข้อความหนึ่งเป็นจริงได้ — "กราฟสองอันนี้ isomorphic กัน" "ฉันรู้รหัสผ่าน" — ผ่านการโต้ตอบไปมาแบบสด ที่บันทึกบทสนทนาของมัน หากถูกอ่านย้อนหลังโดยใครก็ตามที่ไม่ได้อยู่ในการแลกเปลี่ยนนั้น จะพิสูจน์อะไรไม่ได้เลย น้ำหนักเชิงหลักฐานทั้งหมดอยู่ที่การซักถามสด (live interrogation) และความน่าจะเป็นที่ลดลงเรื่อยๆ ของการปล่อยให้คนโกงลอดผ่านไปได้ ไม่ได้อยู่ที่วัตถุนิ่งที่ใครจะตรวจสอบย้อนหลังได้ นั่นแตกหักกับบทพิสูจน์ทุกชิ้นนับตั้งแต่ Euclid เป็นต้นมา ที่บทพิสูจน์คือสิ่งตายตัวที่ยื่นให้ใครคนหนึ่งแล้วเขาตรวจสอบเองตามลำพัง ตรงนี้ปรัชญามีเหตุผลจะคัดค้านได้ว่าการคำนวณนิยาม "บทพิสูจน์" ใหม่ให้กลายเป็นแค่เรื่องเชิงสถิติ และคำตอบที่ตรงไปตรงมาของทฤษฎีความซับซ้อนคือภาพความแน่นอนแบบเก่านั้นไม่เคยมีอยู่จริงในระดับใหญ่ตั้งแต่แรกอยู่แล้ว — บทที่ 12 จะปิดข้อโต้แย้งนี้ให้จบ

Whether "possible in principle but not in practice" earns the name metaphysical depends on whether the boundary is fixed or merely historical. Worst-case complexity classes are mathematical facts, but real instances are not always worst cases: practical SAT solvers now settle satisfiability questions with millions of variables that theory calls intractable, because most real formulas are far from the adversarial examples the proofs are built on. The gap between "P ≠ NP" and "this particular problem, today, on this hardware, is out of reach" is where sociology and engineering quietly do the work that pure complexity theory cannot yet certify.

> "เป็นไปได้ในหลักการแต่ไม่เป็นไปได้ในทางปฏิบัติ" จะสมควรได้ชื่อว่าเป็นหมวดหมู่เชิงอภิปรัชญาหรือไม่ ขึ้นอยู่กับว่าเส้นขอบนั้นตายตัวหรือเป็นแค่เรื่องเชิงประวัติศาสตร์ คลาสความซับซ้อน (complexity class) แบบกรณีเลวร้ายที่สุดเป็นข้อเท็จจริงทางคณิตศาสตร์ แต่ตัวอย่างจริงไม่ได้เป็นกรณีเลวร้ายที่สุดเสมอไป: ตัวแก้ SAT เชิงปฏิบัติในปัจจุบันแก้คำถาม satisfiability ที่มีตัวแปรเป็นล้านตัวได้ ทั้งที่ทฤษฎีเรียกมันว่าแก้ไม่ได้ในทางปฏิบัติ (intractable) เพราะสูตรจริงส่วนใหญ่ห่างไกลจากตัวอย่างปฏิปักษ์ (adversarial examples) ที่บทพิสูจน์เหล่านั้นสร้างขึ้นมาบนฐานของมัน ช่องว่างระหว่าง "P ≠ NP" กับ "ปัญหาข้อนี้โดยเฉพาะ วันนี้ บนฮาร์ดแวร์นี้ อยู่นอกเอื้อม" คือจุดที่สังคมวิทยาและวิศวกรรมเงียบๆ ทำงานที่ทฤษฎีความซับซ้อนล้วนๆ ยังรับรองไม่ได้

Connections · เชื่อมโยง

Chapter 09 established what computation can do at all; this chapter measures what it can do inside a clock. Chapter 03 asked whether machine testimony counts as knowledge; interactive proof is that question with a probability attached. Chapter 06 meets the same impossibility-result shape in algorithmic fairness. Chapter 11 takes up Kolmogorov complexity, the cousin of computational complexity that measures description length rather than running time.

> บทที่ 09 วางรากฐานว่าการคำนวณทำอะไรได้บ้างในหลักการ บทนี้วัดว่ามันทำอะไรได้ภายในเวลาที่มีจำกัด บทที่ 03 ถามว่าคำให้การของเครื่องจักรนับเป็นความรู้หรือไม่ interactive proof คือคำถามเดียวกันนั้นที่ติดความน่าจะเป็นเข้าไปด้วย บทที่ 06 เจอรูปแบบผลลัพธ์ความเป็นไปไม่ได้แบบเดียวกันนี้ในความเป็นธรรมเชิงอัลกอริทึม (algorithmic fairness) บทที่ 11 รับช่วงต่อด้วย Kolmogorov complexity ญาติของความซับซ้อนเชิงคำนวณที่วัดความยาวคำอธิบายแทนที่จะวัดเวลาในการรัน

Open questions · คำถามที่ยังเปิดอยู่

If P vs NP were resolved tomorrow, would it settle a metaphysical question or only a mathematical one? Does the gap between worst-case hardness and real-world tractability mean complexity classes describe the world, or only the adversaries mathematicians have chosen to imagine? And if bounded rationality is a physical necessity for any agent, what is left of the classical picture of a fully rational ideal observer that philosophy built its theories of knowledge and choice around?

> ถ้า P vs NP ถูกคลี่คลายพรุ่งนี้ มันจะยุติคำถามเชิงอภิปรัชญา หรือแค่คำถามเชิงคณิตศาสตร์เท่านั้น? ช่องว่างระหว่างความยากแบบกรณีเลวร้ายที่สุดกับความแก้ได้จริงในโลกจริง หมายความว่าคลาสความซับซ้อนอธิบายโลกจริง หรือแค่อธิบายปฏิปักษ์ที่นักคณิตศาสตร์เลือกจินตนาการขึ้นมาเท่านั้น? และถ้าเหตุผลมีขอบเขต (bounded rationality) เป็นความจำเป็นทางกายภาพสำหรับผู้กระทำใดๆ ก็ตาม เหลืออะไรอยู่บ้างจากภาพคลาสสิกของผู้สังเกตการณ์ในอุดมคติที่มีเหตุผลเต็มเปี่ยม ซึ่งปรัชญาสร้างทฤษฎีความรู้และการเลือกของตัวเองขึ้นมาจากภาพนั้น?

Go deeper · อ่านต่อ

— Scott Aaronson, *Quantum Computing Since Democritus* (Cambridge: Cambridge University Press, 2013).

> Scott Aaronson, Quantum Computing Since Democritus (Cambridge: Cambridge University Press, 2013).

— Stephen Cook, "The Complexity of Theorem-Proving Procedures," *Proceedings of the Third Annual ACM Symposium on Theory of Computing* (1971): 151–158; Scott Aaronson, "Why Philosophers Should Care About Computational Complexity" (2011) — arxiv.org/abs/1108.1791.

> Stephen Cook, "The Complexity of Theorem-Proving Procedures," Proceedings of the Third Annual ACM Symposium on Theory of Computing (1971): 151–158; Scott Aaronson, "Why Philosophers Should Care About Computational Complexity" (2011) — arxiv.org/abs/1108.1791.

— Stanford Encyclopedia of Philosophy, "Computational Complexity Theory" — plato.stanford.edu/entries/computational-complexity.

> Stanford Encyclopedia of Philosophy, "Computational Complexity Theory" — plato.stanford.edu/entries/computational-complexity.

CHAPTER 11

Information

สารสนเทศ

Third hearing of Part II. Chapters 09 and 10 measured what computation can do and how long it takes; this one asks what computation actually moves when it runs at all — and whether that thing is the whole world, or a very useful abstraction from it.

การไต่สวนครั้งที่สามของภาค II บทที่ 09 และ 10 วัดว่าการคำนวณทำอะไรได้บ้างและใช้เวลานานแค่ไหน บทนี้ถามว่าการคำนวณเคลื่อนอะไรจริงๆ เมื่อมันทำงาน — และสิ่งนั้นคือโลกทั้งใบหรือเป็นแค่ abstraction ที่มีประโยชน์มากที่สกัดออกมาจากโลก

Is information the stuff the world is made of?

สารสนเทศคือสิ่งที่โลกถูกสร้างขึ้นมาจากมันหรือไม่?

Claude Shannon opened "A Mathematical Theory of Communication" in the *Bell System Technical Journal* in July and October 1948 with a sentence that would outlive the telephone-switching problem it was written to solve: "Frequently the messages have meaning; that is, they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem." Chapter 00 quoted the line to mark computing's original forgetting; this chapter audits what the bracket actually bought, and what it cost. What Shannon needed was a way to quantify how much a signal could be compressed and how reliably it could survive noise, regardless of what it was about — and his measure, entropy, $H = -\Sigma p \log p$, does exactly that. A page of Shakespeare and a page of the same length generated by a fair coin flip can carry identical Shannon information if their symbol frequencies match, because the quantity is about surprise, not significance. The bracket was honest and correct for the problem in front of him. It has been treated

as a metaphysics ever since — a discipline that can measure information without asking what it is *of* eventually builds systems it can quantify and cannot interpret, and that gap is this chapter's subject.

> Claude Shannon เปิดบทความ "A Mathematical Theory of Communication" ใน Bell System Technical Journal ช่วงเดือนกรกฎาคมและตุลาคม 1948 ด้วยประโยคที่จะอยู่ยั้งยืนยงเกินกว่าปัญหาการสลับสายโทรศัพท์ที่มันถูกเขียนขึ้นมาเพื่อแก้: "บ่อยครั้งที่ข้อความมีความหมาย นั่นคือ มันอ้างถึงหรือมีความสัมพันธ์กับสิ่งที่มีอยู่จริงทางกายภาพหรือทางมโนทัศน์บางระบบ แง่มุมเชิงความหมาย (semantic) เหล่านี้ของการสื่อสารไม่เกี่ยวข้องกับปัญหาเชิงวิศวกรรม" บทที่ 00 อ้างประโยคนี้เพื่อทำเครื่องหมายการลืมดั้งเดิมของวงการคำนวณ บทนี้ตรวจสอบว่าวงเล็บนั้นซื้ออะไรมาได้จริงๆ และต้องจ่ายอะไรไปเป็นการแลกเปลี่ยน สิ่งที่ Shannon ต้องการคือวิธีวัดปริมาณว่าสัญญาณหนึ่งบีบอัดได้มากแค่ไหน และมันจะรอดจากสัญญาณรบกวนได้น่าเชื่อถือแค่ไหน โดยไม่สนว่ามันพูดถึงเรื่องอะไร — และหน่วยวัดของเขาเอนโทรปี (entropy) H = −Σp log p ทำแบบนั้นได้เป๊ะ หน้ากระดาษจากงานของ Shakespeare กับหน้ากระดาษความยาวเท่ากันที่สร้างจากการโยนเหรียญที่ยุติธรรมสามารถมีสารสนเทศแบบ Shannon เท่ากันได้ ถ้าความถี่ของสัญลักษณ์ตรงกัน เพราะปริมาณนี้วัดความประหลาดใจ ไม่ใช่ความสำคัญ วงเล็บนั้นซื่อตรงและถูกต้องสำหรับปัญหาที่อยู่ตรงหน้าเขา แต่นับแต่นั้นมันถูกมองเป็นอภิปรัชญาไปเสียอย่างนั้น — สาขาวิชาที่วัดสารสนเทศได้โดยไม่ต้องถามว่ามันเป็นสารสนเทศของอะไร ในที่สุดก็สร้างระบบที่มันวัดปริมาณได้แต่ตีความไม่ได้ และช่องว่างนั้นคือหัวข้อของบทนี้

Shannon's entropy measures a message against a known probability distribution of *possible* messages; it says nothing about a single object's own intrinsic structure. Andrei Kolmogorov's "Three Approaches to the Quantitative Definition of Information" (*Problems of Information Transmission* 1, no. 1, 1965) supplied that: the complexity of a string is the length of the shortest program that outputs it and stops. A million zeros compresses to a short program ("print 0, one million times"); a string of the same length generated by fair coin flips compresses, with overwhelming probability, to nothing shorter than itself — a formal definition of randomness as incompressibility. Kolmogorov was not first. Ray Solomonoff had described algorithmic probability independently at Caltech in 1960 and published "A

Formal Theory of Inductive Inference" (*Information and Control* 7, 1964: 1–22, 224–254) for a different purpose entirely: not measuring randomness but formalizing induction itself. Solomonoff's universal prior weights every hypothesis by the length of the shortest program that generates the observed data, turning "prefer the simplest explanation" from a maxim into an equation — Occam's razor with a ruler. Solomonoff's work went almost unnoticed until Kolmogorov's later, more rigorously presented papers cited it; priority runs one way, fame ran the other, and Jorma Rissanen's minimum description length principle (1978) turned the same idea into a working tool for choosing between statistical models.

> เอนโทรปีของ Shannon วัดข้อความหนึ่งเทียบกับการแจกแจงความน่าจะเป็นที่รู้อยู่แล้วของข้อความที่เป็นไปได้ทั้งหมด มันไม่บอกอะไรเกี่ยวกับโครงสร้างในตัวของวัตถุชิ้นเดียวเลย บทความ "Three Approaches to the Quantitative Definition of Information" ของ Andrei Kolmogorov (Problems of Information Transmission 1, no. 1, 1965) ให้คำตอบนั้นมา: ความซับซ้อนโคลโมโกรอฟ (Kolmogorov complexity) ของสตริงหนึ่งคือความยาวของโปรแกรมที่สั้นที่สุดที่ให้ผลลัพธ์เป็นสตริงนั้นแล้วหยุด เลขศูนย์หนึ่งล้านตัวบีบอัดเหลือเป็นโปรแกรมสั้นๆ ได้ ("พิมพ์ 0 หนึ่งล้านครั้ง") ส่วนสตริงความยาวเท่ากันที่สร้างจากการโยนเหรียญที่ยุติธรรม แทบทุกครั้งจะบีบอัดเหลือสั้นกว่าตัวมันเองไม่ได้เลย — นิยามเชิงรูปนัยของความสุ่มในฐานะการบีบอัดไม่ได้ (incompressibility) Kolmogorov ไม่ใช่คนแรก Ray Solomonoff อธิบายความน่าจะเป็นเชิงอัลกอริทึม (algorithmic probability) ไว้อย่างอิสระที่ Caltech ตั้งแต่ปี 1960 และตีพิมพ์ "A Formal Theory of Inductive Inference" (Information and Control 7, 1964: 1–22, 224–254) ด้วยจุดประสงค์ที่ต่างออกไปโดยสิ้นเชิง: ไม่ใช่วัดความสุ่ม แต่ทำให้การอุปนัยเองเป็นรูปนัย prior สากล (universal prior) ของ Solomonoff ให้น้ำหนักสมมติฐานทุกข้อตามความยาวของโปรแกรมที่สั้นที่สุดที่สร้างข้อมูลที่สังเกตได้ พลิก "เลือกคำอธิบายที่ง่ายที่สุด" จากคติพจน์ให้กลายเป็นสมการ — มีดโกนของ Occam ที่มีไม้บรรทัดกำกับ งานของ Solomonoff แทบไม่มีใครสนใจ จนกระทั่งบทความของ Kolmogorov ที่เขียนขึ้นทีหลังและนำเสนออย่างเข้มงวดกว่าอ้างอิงถึงมัน ความเป็นเจ้าของก่อนวิ่งไปทาง

> หนึ่ง ชื่อเสียงวิ่งไปอีกทาง และหลักความยาวคำอธิบายน้อยที่สุด (minimum description length, MDL) ของ Jorma Rissanen ปี 1978 พลิกไอเดียเดียวกันนี้ให้กลายเป็นเครื่องมือใช้งานได้จริงสำหรับเลือกระหว่างโมเดลทางสถิติ

Rolf Landauer supplied the fact that keeps information from floating free of physics altogether. "Irreversibility and Heat Generation in the Computing Process" (*IBM Journal of Research and Development* 5, 1961: 183–191) showed that erasing one bit of memory — a logically irreversible operation, since the bit's prior state cannot be recovered from its blank successor — necessarily dissipates a minimum quantity of heat, on the order of kT ln 2. The claim sat as a theoretical curiosity for half a century until Antoine Bérut and colleagues built a single colloidal particle in a bistable potential and measured the predicted heat directly, published as "Experimental Verification of Landauer's Principle Linking Information and Thermodynamics" (*Nature* 483, 2012: 187–189). Landauer's own slogan, "information is physical," is not a metaphor: a bit erased anywhere in the universe has a thermodynamic price tag, which is exactly why Maxwell's demon — the imagined creature that seems to violate the second law of thermodynamics by sorting fast molecules from slow ones for free — turns out to owe its debt not at the sorting but at the moment it must erase its own memory of which molecule went where.

> Rolf Landauer ให้ข้อเท็จจริงที่กันไม่ให้สารสนเทศลอยตัวเป็นอิสระจากฟิสิกส์ไปเสียทีเดียว ด้วยสิ่งที่ภายหลังถูกเรียกว่าหลักการแลนเดาเออร์ (Landauer's principle) บทความ "Irreversibility and Heat Generation in the Computing Process" (IBM Journal of Research and Development 5, 1961: 183–191) แสดงว่าการลบความจำหนึ่งบิต — ปฏิบัติการที่ย้อนกลับไม่ได้เชิงตรรกะ เพราะสถานะก่อนหน้าของบิตนั้นกู้คืนจากสถานะว่างที่ตามมาไม่ได้ — จำเป็นต้องปล่อยความร้อนออกมาอย่างน้อยที่สุดปริมาณหนึ่ง ในระดับ kT ln 2 ข้อกล่าวอ้างนี้เป็นแค่เรื่องน่าสนใจเชิงทฤษฎีอยู่นานครึ่งศตวรรษ จนกระทั่ง Antoine Bérut กับทีมสร้างอนุภาคคอลลอยด์เดี่ยวในศักย์แบบสองสถานะ (bistable potential) แล้ววัดความร้อนที่ทำนายไว้ได้โดยตรง ตีพิมพ์เป็น "Experimental Verification of Landauer's Principle Linking Information and Thermodynamics" (Nature 483, 2012:

> 187–189) คำขวัญของ Landauer เอง "สารสนเทศคือสิ่งเชิงกายภาพ" ไม่ใช่คำอุปมา: บิตหนึ่งที่ถูกลบไม่ว่าจะที่ไหนในเอกภพมีป้ายราคาทางเทอร์โมไดนามิกส์กำกับอยู่ นั่นคือเหตุผลว่าทำไมปีศาจของ Maxwell — สิ่งมีชีวิตในจินตนาการที่ดูเหมือนจะละเมิดกฎข้อที่สองของเทอร์โมไดนามิกส์ด้วยการแยกโมเลกุลเร็วออกจากโมเลกุลช้าโดยไม่เสียค่าใช้จ่าย — กลับกลายเป็นว่าติดหนี้อยู่ ไม่ใช่ตอนที่แยก แต่ตอนที่มันต้องลบความจำของตัวเองว่าโมเลกุลไหนไปทางไหน

John Archibald Wheeler took the physical entanglement of information and pushed it all the way to ontology. "Information, Physics, Quantum: The Search for Links," delivered at a Santa Fe Institute meeting in 1989 and published the following year, proposed that "every *it* — every particle, every field of force, even the spacetime continuum itself — derives its function, its meaning, its very existence entirely" from "the apparatus-elicited answers to yes-or-no questions, binary choices, bits." Existence, on this reading, is downstream of measurement. Luciano Floridi's philosophy of information generalizes the same instinct into a proposed successor discipline to metaphysics, built on the claim that reality is best described at "levels of abstraction" defined by the observables a system tracks — informational structure, not stuff, as what is ultimately real.

> John Archibald Wheeler เอาความพัวพันเชิงกายภาพของสารสนเทศ ผลักดันมันไปจนถึงภววิทยา (ontology) บทความ "Information, Physics, Quantum: The Search for Links" ที่บรรยายในงานประชุมของ Santa Fe Institute ปี 1989 และตีพิมพ์ปีถัดมา เสนอว่า "ทุก it — อนุภาคทุกตัว สนามแรงทุกสนาม แม้แต่ตัวความต่อเนื่องของกาลอวกาศเอง — ได้รับหน้าที่ ความหมาย และการมีอยู่ของตัวมันทั้งหมด" มาจาก "คำตอบที่อุปกรณ์ดึงออกมาจากคำถามใช่หรือไม่ใช่ ตัวเลือกฐานสอง บิต" การมีอยู่ ตามการอ่านนี้ จึงเป็นสิ่งที่ตามมาทีหลังการวัด ปรัชญาสารสนเทศของ Luciano Floridi ขยายสัญชาตญาณเดียวกันนี้ให้กลายเป็นสาขาที่เสนอตัวขึ้นมาแทนอภิปรัชญา สร้างขึ้นบนข้อกล่าวอ้างที่ว่าความเป็นจริงอธิบายได้ดีที่สุดที่ "ระดับของ abstraction" ที่นิยามโดยสิ่งที่สังเกตได้ (observables) ซึ่งระบบหนึ่งติดตาม — โครงสร้างเชิงสารสนเทศ ไม่ใช่วัตถุ คือสิ่งที่เป็นจริงในท้ายที่สุด

Here the examination turns, because a fact about physical cost is not the same claim as a fact about physical constitution, and philosophy is entitled to press exactly that distinction. Landauer proved that erasing information costs energy; he did not thereby prove that energy, mass, and spacetime are *made of* information rather than merely correlated with it whenever computation happens inside them. Floridi's own framework quietly concedes the point by making "information" relative to a chosen level of abstraction — a physicist's level and a biologist's level carve the same event into different informational units, which is difficult to square with information being the one substrate underneath both. Wheeler's participatory universe remains a research program rather than an established result, and most physicists who work on quantum foundations treat "it from bit" as a suggestive slogan, not a settled dependency. The identification overreaches exactly where Shannon's original bracket was most honest: a theory that measures information while setting meaning aside cannot, without further argument, turn around and claim that meaning was never there to set aside.

> ตรงนี้เองที่การซักถามหันเหทิศทาง เพราะข้อเท็จจริงเกี่ยวกับต้นทุนเชิงกายภาพไม่ใช่ข้อกล่าวอ้างเดียวกับข้อเท็จจริงเกี่ยวกับองค์ประกอบเชิงกายภาพ และปรัชญามีสิทธิ์กดดันให้เห็นข้อแบ่งแยกนั้นเป๊ะๆ Landauer พิสูจน์ว่าการลบสารสนเทศมีต้นทุนเป็นพลังงาน แต่เขาไม่ได้พิสูจน์ไปด้วยว่าพลังงาน มวล และกาลอวกาศ สร้างขึ้นจาก สารสนเทศ แทนที่จะแค่มีความสัมพันธ์กับมันทุกครั้งที่การคำนวณเกิดขึ้นข้างในมัน กรอบคิดของ Floridi เองแอบยอมรับประเด็นนี้อยู่กลายๆ ด้วยการทำให้ "สารสนเทศ" สัมพัทธ์กับระดับของ abstraction ที่เลือกไว้ — ระดับของนักฟิสิกส์กับระดับของนักชีววิทยาตัดเหตุการณ์เดียวกันออกเป็นหน่วยสารสนเทศที่ต่างกัน ซึ่งยากจะปรับให้เข้ากับการที่สารสนเทศเป็นสารตั้งต้นหนึ่งเดียวที่อยู่ใต้ทั้งสองระดับนั้น เอกภพเชิงมีส่วนร่วม (participatory universe) ของ Wheeler ยังคงเป็นแค่โครงการวิจัย ไม่ใช่ผลลัพธ์ที่ยืนยันแล้ว และนักฟิสิกส์ส่วนใหญ่ที่ทำงานด้านรากฐานควอนตัมมอง it from bit (สสารจากบิต) เป็นแค่คำขวัญที่ชวนคิด ไม่ใช่การพึ่งพาที่ยุติแล้ว การระบุตัวตนแบบนี้ยื่นเกินไปพอดีตรงจุดที่วงเล็บดั้งเดิมของ Shannon ซื่อตรงที่สุด: ทฤษฎีที่วัดสารสนเทศได้โดยกันความหมายออกไป จะหันกลับมาอ้างโดยไม่มีข้อโต้แย้งเพิ่มเติมว่าความหมายไม่เคยมีอยู่ให้ต้องกันออกไปตั้งแต่แรกไม่ได้

Connections · เชื่อมโยง

Chapter 00 quoted Shannon's bracket to mark computing's forgetting; this chapter audits it in full. Chapter 01 keeps to Shannon's earlier, narrower 1937 circuit work. Chapter 02 takes up Floridi's informational realism as a question about what a program is. Chapter 10 supplies the sibling notion — computational complexity measures time, Kolmogorov complexity measures description length, and the two rarely coincide. Chapter 15 takes digital physics — it-from-bit's descendants, from Zuse to Wolfram — to the question of whether the universe itself computes.

> บทที่ 00 อ้างวงเล็บของ Shannon เพื่อทำเครื่องหมายการลืมของวงการคำนวณ บทนี้ตรวจสอบมันอย่างเต็มที่ บทที่ 01 อยู่กับงานวงจรปี 1937 ของ Shannon ที่แคบกว่าและเก่ากว่า บทที่ 02 หยิบสัจนิยมเชิงสารสนเทศ (informational realism) ของ Floridi มาเป็นคำถามว่าโปรแกรมหนึ่งคืออะไร บทที่ 10 ให้แนวคิดพี่น้องกัน — ความซับซ้อนเชิงคำนวณวัดเวลา ส่วน Kolmogorov complexity วัดความยาวคำอธิบาย ทั้งสองแทบไม่เคยตรงกัน บทที่ 15 พาฟิสิกส์เชิงดิจิทัล (digital physics) — ทายาทของ it from bit ตั้งแต่ Zuse ถึง Wolfram — ไปสู่คำถามว่าเอกภพเองคำนวณหรือไม่

Open questions · คำถามที่ยังเปิดอยู่

If Kolmogorov complexity is uncomputable in general — no algorithm can determine the shortest program for an arbitrary string — what does that limit say about how far Occam's razor can ever be mechanized? Does Landauer's principle really license "information is physical," or only "erasing information has a physical cost," and is the difference between those two claims the whole of the dispute over digital physics? And if levels of abstraction are chosen by observers rather than discovered in nature, can philosophy of information still claim to be describing what is real rather than what is useful to track?

> ถ้าความซับซ้อนโคลโมโกรอฟคำนวณไม่ได้โดยทั่วไป — ไม่มีอัลกอริทึมใดหาโปรแกรมที่สั้นที่สุดสำหรับสตริงใดๆ ได้ — ขีดจำกัดนั้นบอกอะไรเกี่ยวกับว่ามีดโกนของ Occam จะถูกทำให้เป็นกลไกได้ไกลแค่ไหน? หลักการแลนเดาเออร์อนุญาตให้

> พูดว่า "สารสนเทศคือสิ่งเชิงกายภาพ" ได้จริงหรือ หรืออนุญาตแค่ "การลบสารสนเทศมีต้นทุนเชิงกายภาพ" เท่านั้น และความต่างระหว่างสองข้อกล่าวอ้างนี้คือข้อพิพาททั้งหมดเรื่องฟิสิกส์เชิงดิจิทัลใช่หรือไม่? และถ้าระดับของ abstraction ถูกเลือกโดยผู้สังเกตการณ์ แทนที่จะถูกค้นพบในธรรมชาติ ปรัชญาสารสนเทศยังอ้างได้อยู่ไหมว่ากำลังอธิบายสิ่งที่เป็นจริง ไม่ใช่แค่สิ่งที่มีประโยชน์จะติดตาม?

Go deeper · อ่านต่อ

— James Gleick, *The Information: A History, a Theory, a Flood* (New York: Pantheon, 2011).

> James Gleick, The Information: A History, a Theory, a Flood (New York: Pantheon, 2011).

— Claude Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal* 27 (1948): 379–423, 623–656; Ray Solomonoff, "A Formal Theory of Inductive Inference," *Information and Control* 7 (1964): 1–22, 224–254.

> Claude Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal 27 (1948): 379–423, 623–656; Ray Solomonoff, "A Formal Theory of Inductive Inference," Information and Control 7 (1964): 1–22, 224–254.

— Stanford Encyclopedia of Philosophy, "Information" — plato.stanford.edu/entries/information.

> Stanford Encyclopedia of Philosophy, "Information" — plato.stanford.edu/entries/information.

CHAPTER 12

Types and Proofs

ชนิดกับบทพิสูจน์

Fourth hearing of Part II. The last three chapters asked what computation can do, how long it takes, and what it moves; this one asks what happens to the oldest philosophical prize — certainty — once proof itself becomes a checkable artifact.

การไต่สวนครั้งที่สี่ของภาค II สามบทที่ผ่านมาถามว่าการคำนวณทำอะไรได้บ้าง ใช้เวลานานแค่ไหน และเคลื่อนอะไร บทนี้ถามว่าเกิดอะไรขึ้นกับรางวัลเชิงปรัชญาที่เก่าแก่ที่สุด — ความแน่นอน — เมื่อบทพิสูจน์เองกลายเป็นสิ่งประดิษฐ์ที่ตรวจสอบได้

What happens to certainty when proof becomes an artifact?

เกิดอะไรขึ้นกับความแน่นอน เมื่อบทพิสูจน์กลายเป็นสิ่งประดิษฐ์?

In 1976 Kenneth Appel and Wolfgang Haken proved the four-color theorem with the help of a computer program that checked close to fifteen hundred map configurations no human being would ever survey by hand — chapter 03 already sat with the shock that verdict caused: a proof too long for any mathematician to read start to finish. Nearly thirty years later Georges Gonthier closed a different case about the same theorem. His formalization, completed in the Coq proof assistant and published as "Formal Proof — The Four-Color Theorem" (*Notices of the American Mathematical Society* 55, 2008: 1382–1393), did not merely rerun Appel and Haken's argument; it re-derived the whole proof, combinatorics and case-checking alike, as a sequence of steps validated by a small trusted kernel that a single person can, in principle, audit line by line. The theorem that once symbolized proof out-

running human comprehension became, in Gonthier's hands, the theorem that showed comprehension could be replaced by something more exacting than any human reader: a machine that does not skim.

> ปี 1976 Kenneth Appel และ Wolfgang Haken พิสูจน์ทฤษฎีบทสี่สี (four-color theorem) ได้ด้วยความช่วยเหลือของโปรแกรมคอมพิวเตอร์ที่ตรวจสอบการจัดวางแผนที่เกือบหนึ่งพันห้าร้อยแบบ ซึ่งไม่มีมนุษย์คนไหนจะสำรวจด้วยมือได้เลย — บทที่ 03 นั่งอยู่กับแรงสั่นสะเทือนที่คำตัดสินนั้นก่อไว้แล้ว: บทพิสูจน์ที่ยาวเกินกว่านักคณิตศาสตร์คนใดจะอ่านจบตั้งแต่ต้นจนจบได้ เกือบสามสิบปีต่อมา Georges Gonthier ปิดคดีอีกคดีหนึ่งเกี่ยวกับทฤษฎีบทเดียวกันนี้ การทำให้เป็นรูปนัย (formalization) ของเขา ที่ทำเสร็จใน proof assistant ชื่อ Coq และตีพิมพ์เป็น "Formal Proof — The Four-Color Theorem" (Notices of the American Mathematical Society 55, 2008: 1382–1393) ไม่ได้แค่รับข้อโต้แย้งของ Appel กับ Haken ซ้ำ แต่มันสร้างบทพิสูจน์ทั้งชิ้นขึ้นมาใหม่ ทั้ง combinatorics และการตรวจสอบกรณีต่างๆ ในฐานะลำดับขั้นตอนที่ยืนยันความถูกต้องโดย kernel (แกนตรวจสอบ) เล็กๆ ที่ไว้ใจได้ ซึ่งคนคนเดียวตรวจสอบได้ทีละบรรทัดในหลักการ ทฤษฎีบทที่ครั้งหนึ่งเคยเป็นสัญลักษณ์ของบทพิสูจน์ที่วิ่งเกินความเข้าใจของมนุษย์ในมือของ Gonthier กลับกลายเป็นทฤษฎีบทที่แสดงว่าความเข้าใจถูกแทนที่ได้ด้วยบางสิ่งที่เข้มงวดกว่าผู้อ่านมนุษย์คนใด: เครื่องจักรที่ไม่มีวันอ่านผ่านๆ

What makes Gonthier's kernel trustworthy is a correspondence discovered, not designed, twice over. Haskell Curry noticed in 1934 that the types assigned to combinators in combinatory logic could be read as axiom schemes of intuitionistic implicational logic, and in 1958 that a fragment of Hilbert-style proof systems coincided with a fragment of typed combinatory logic. William Howard made the analogy exact three decades later without knowing Curry's full history, circulating a manuscript in 1969 that was finally published as "The Formulae-as-Types Notion of Construction" in a 1980 festschrift for Curry edited by Jonathan Seldin and J. Roger Hindley. Howard showed that a proof, written in the natural-deduction style Gerhard Gentzen had introduced in 1935, and a program, written in the simply typed lambda calculus, are the same object viewed from two directions: a proposition is a type, a proof of it is a program of that type, and simplifying

a proof — removing a detour — is executing the program. N. G. de Bruijn arrived at a version of the same idea independently at almost the same time, building it into his Automath system before either he or Howard knew of the other's work. Three researchers, working from logic, from combinators, and from an automated proof-checker, kept finding the identical seam — the strongest kind of evidence this book has for its own claim that philosophy and computing are one project rediscovered rather than two projects borrowed from each other.

> สิ่งที่ทำให้ kernel ของ Gonthier ไว้ใจได้คือความสอดคล้อง (correspondence) ที่ถูก "ค้นพบ" ไม่ใช่ "ออกแบบ" ซ้ำกันถึงสองรอบ Haskell Curry สังเกตในปี 1934 ว่าชนิด (type) ที่กำหนดให้ combinator ในตรรกะเชิงการจัด (combinatory logic) อ่านได้ในฐานะ axiom scheme ของตรรกะเชิงเงื่อนไขแบบสัญชานิยม (intuitionistic) และในปี 1958 ว่าส่วนหนึ่งของระบบพิสูจน์แบบ Hilbert ตรงกับส่วนหนึ่งของตรรกะเชิงการจัดที่มีชนิดกำกับ William Howard ทำให้การเทียบเคียงนี้แม่นยำขึ้นสามทศวรรษให้หลังโดยไม่รู้ประวัติทั้งหมดของ Curry เผยแพร่ต้นฉบับในปี 1969 ซึ่งสุดท้ายตีพิมพ์เป็น "The Formulae-as-Types Notion of Construction" ใน festschrift ปี 1980 ที่ทำขึ้นเพื่อ Curry โดยมี Jonathan Seldin และ J. Roger Hindley เป็นบรรณาธิการ Howard แสดงว่าบทพิสูจน์หนึ่ง ที่เขียนในสไตล์การนิรนัยธรรมชาติ (natural deduction) ที่ Gerhard Gentzen แนะนำไว้ในปี 1935 กับโปรแกรมหนึ่ง ที่เขียนในแคลคูลัสแลมบ์ดาที่มีชนิดอย่างง่าย (simply typed lambda calculus) เป็นวัตถุเดียวกันที่มองจากสองทิศทาง: ประพจน์หนึ่งคือชนิดหนึ่ง บทพิสูจน์ของมันคือโปรแกรมของชนิดนั้น และการทำให้บทพิสูจน์ง่ายขึ้น — การตัดทางอ้อมทิ้ง — ก็คือการรันโปรแกรมนั่นเอง N. G. de Bruijn มาถึงไอเดียแบบเดียวกันนี้อย่างอิสระในช่วงเวลาใกล้เคียงกันแทบจะพอดี สร้างมันเข้าไปในระบบ Automath ของเขาก่อนที่ทั้งเขาและ Howard จะรู้ว่าอีกฝ่ายทำงานเรื่องเดียวกันอยู่ นักวิจัยสามคน ทำงานจากตรรกศาสตร์ จาก combinator และจากตัวตรวจสอบบทพิสูจน์อัตโนมัติ ต่างพบรอยต่อเดียวกันนี้ซ้ำแล้วซ้ำเล่า — หลักฐานที่หนักแน่นที่สุดที่หนังสือเล่มนี้มีสำหรับข้อกล่าวอ้างของตัวมันเองที่ว่าปรัชญากับการคำนวณเป็นโครงการเดียวที่ถูกค้นพบซ้ำ ไม่ใช่สองโครงการที่ยืมกัน

The correspondence gave a century-old heresy its revenge. L. E. J. Brouwer's 1908 paper "The Unreliability of the Logical Principles" rejected the law of excluded middle — the claim that every statement is either true or false — for mathematics whose objects are mental constructions rather than mind-independent facts, building on the intuitionistic program he had opened in his 1907 dissertation. For most of the twentieth century this looked like a minority taste, a scruple most working mathematicians could safely ignore. Curry–Howard explains why it stopped being optional inside a proof assistant: a constructive proof that something exists must, under the correspondence, be a program that builds the thing, while a classical proof by contradiction — assume no such object exists, derive a contradiction, conclude one does — corresponds to no program at all, because it never says how to build anything. Coq, Agda, and Lean are, at their foundations, implementations of intuitionistic type theory not out of philosophical loyalty to Brouwer but because only the constructive fragment hands back code you can run. Intuitionism became load-bearing engineering exactly at the point where it stopped being asked to defend itself philosophically.

> ความสอดคล้องนี้ให้การแก้แค้นแก่ความนอกรีตที่มีอายุนับศตวรรษ บทความปี 1908 ของ L. E. J. Brouwer เรื่อง "The Unreliability of the Logical Principles" ปฏิเสธกฎแห่งไตรวินิจฉัย (law of excluded middle) — ข้อกล่าวอ้างที่ว่าทุกประพจน์เป็นจริงหรือเท็จอย่างใดอย่างหนึ่งเสมอ — สำหรับคณิตศาสตร์ที่วัตถุของมันเป็นสิ่งสร้างทางจิต ไม่ใช่ข้อเท็จจริงที่เป็นอิสระจากจิต โดยต่อยอดจากโครงการสัญชานนิยม (intuitionism) ที่เขาเปิดไว้ในวิทยานิพนธ์ปี 1907 ตลอดศตวรรษที่ 20 ส่วนใหญ่ นี่ดูเหมือนรสนิยมของคนกลุ่มน้อย ความลังเลใจที่นักคณิตศาสตร์ทำงานส่วนใหญ่มองข้ามได้อย่างปลอดภัย Curry–Howard อธิบายว่าทำไมมันถึงเลิกเป็นทางเลือกได้ในโลกของ proof assistant: บทพิสูจน์เชิงสร้าง (constructive proof) ที่ว่าบางสิ่งมีอยู่จริง ภายใต้ความสอดคล้องนี้ ต้องเป็นโปรแกรมที่สร้างสิ่งนั้นขึ้นมา ในขณะที่บทพิสูจน์เชิงคลาสสิก (classical proof) แบบพิสูจน์โดยข้อขัดแย้ง — สมมติว่าไม่มีวัตถุแบบนั้นอยู่ อนุมานความขัดแย้งออกมา แล้วสรุปว่ามันมีอยู่ — ไม่สอดคล้องกับโปรแกรมใดเลย เพราะมันไม่เคยบอกว่าจะสร้างอะไรขึ้นมาได้อย่างไร Coq, Agda และ Lean ที่รากฐานของมันคือการสร้าง (implementation) ของทฤษฎีชนิดสัญชานนิยม (intuitionistic type theory) ไม่ใช่เพราะความภักดีทางปรัชญา

> ต่อ Brouwer แต่เพราะมีแต่ส่วนที่เป็นเชิงสร้างเท่านั้นที่คืนโค้ดที่รันได้จริงกลับมาให้ สัญชานิยมกลายเป็นวิศวกรรมที่แบกรับน้ำหนักจริง พอดีตรงจุดที่มันเลิกถูกขอให้ปกป้องตัวเองเชิงปรัชญา

Once that machinery existed, mathematics started handing it its hardest, longest proofs. Thomas Hales completed a proof of the Kepler conjecture — that no packing of equal spheres is denser than the familiar grocer's stack — in 1998, but the 250-page argument, part conventional and part computer-assisted, took twelve referees four years and left them only 99 percent confident it was correct. Hales responded in 2003 by launching the Flyspeck project to re-derive every step inside HOL Light and Isabelle; the formalization was completed in 2014, checked and independently rechecked that year, and published as "A Formal Proof of the Kepler Conjecture" (*Forum of Mathematics, Pi*, 2017). Flyspeck's tens of thousands of lemmas rest on a kernel small enough for a handful of people to inspect completely, in place of twelve referees who could only sample a proof too long to read whole. Lean's mathlib, a continuously growing, community-verified library built on the same foundations, now formalizes results across the whole of mathematics as they are proved, turning "has this been checked" into a question with a machine-checkable answer rather than a matter of reputation.

> พอกลไกนั้นมีอยู่แล้ว คณิตศาสตร์ก็เริ่มยื่นบทพิสูจน์ที่ยากที่สุดและยาวที่สุดของมันให้ Thomas Hales พิสูจน์ข้อคาดการณ์ของ Kepler (Kepler conjecture) เสร็จ — ว่าการอัดทรงกลมขนาดเท่ากันไม่มีแบบไหนแน่นกว่ากองแบบที่พ่อค้าขายผลไม้ทำ — ในปี 1998 แต่ข้อโต้แย้ง 250 หน้าซึ่งครึ่งหนึ่งเป็นแบบดั้งเดิมและอีกครึ่งใช้คอมพิวเตอร์ช่วย ใช้เวลาผู้ตรวจทานสิบสองคนถึงสี่ปี และทิ้งพวกเขาไว้กับความมั่นใจแค่ 99 เปอร์เซ็นต์ว่ามันถูกต้อง Hales ตอบโต้ในปี 2003 ด้วยการเปิดโครงการ Flyspeck เพื่อสร้างทุกขั้นตอนใหม่ภายใน HOL Light และ Isabelle การทำให้เป็นรูปนัยเสร็จสมบูรณ์ในปี 2014 ถูกตรวจสอบและตรวจสอบซ้ำอย่างอิสระในปีเดียวกันนั้น แล้วตีพิมพ์เป็น "A Formal Proof of the Kepler Conjecture" (Forum of Mathematics, Pi, 2017) บทตั้ง (lemma) หลายหมื่นข้อของ Flyspeck วางอยู่บน kernel ที่เล็กพอให้คนหยิบมือหนึ่งตรวจสอบได้ครบถ้วน แทนที่จะเป็นผู้ตรวจทานสิบสองคนที่สุ่มตรวจได้แค่บางส่วนของบทพิสูจน์ที่ยาวเกินกว่าจะอ่านจบทั้งหมด

> mathlib ของ Lean ซึ่งเป็นคลังที่โตต่อเนื่องและตรวจสอบยืนยันโดยชุมชน สร้างขึ้นบนรากฐานเดียวกันนี้ ตอนนี้ทำให้ผลลัพธ์ทั่วทั้งคณิตศาสตร์เป็นรูปนัยไปพร้อมกับที่มันถูกพิสูจน์ พลิก "เรื่องนี้ถูกตรวจสอบแล้วหรือยัง" จากเรื่องของชื่อเสียงให้กลายเป็นคำถามที่มีคำตอบตรวจสอบได้ด้วยเครื่องจักร

Vladimir Voevodsky pushed the foundations themselves one step further. *Homotopy Type Theory: Univalent Foundations of Mathematics* — the collective volume written by the Institute for Advanced Study program he led, published in 2013 — proposed reading types not as bare sets but as spaces, and proofs of equality between elements as paths between points — under Voevodsky's univalence axiom, structures that are isomorphic can be treated as identical, which gives mathematics's oldest and most abused word, "equal," a precise, checkable meaning it had never quite had before.

> Vladimir Voevodsky ผลักดันรากฐานเองไปอีกขั้นหนึ่ง Homotopy Type Theory: Univalent Foundations of Mathematics — เล่มรวมที่เขียนโดยโปรแกรมของ Institute for Advanced Study ที่เขาเป็นผู้นำ ตีพิมพ์ในปี 2013 — เสนอให้อ่านชนิด (type) ไม่ใช่ในฐานะเซตเปล่าๆ แต่ในฐานะปริภูมิ (space) และบทพิสูจน์ความเท่ากันระหว่างสมาชิกในฐานะเส้นทางระหว่างจุด — ภายใต้สัจพจน์เอกค่า (univalence axiom) ของ Voevodsky โครงสร้างที่ isomorphic กันสามารถถูกมองว่าเหมือนกันได้ ซึ่งให้ความหมายที่แม่นยำและตรวจสอบได้แก่คำว่า "เท่ากัน" ซึ่งเป็นคำที่เก่าแก่ที่สุดและถูกใช้ผิดมากที่สุดคำหนึ่งในคณิตศาสตร์ — ความหมายที่มันไม่เคยมีมาก่อนอย่างแท้จริง

None of this delivers the certainty Hilbert wanted in 1930. A proof assistant's guarantee is only as strong as its kernel, and the kernel rests on a compiler, which rests on hardware, which can fail — the same regress Wittgenstein pressed against any rule that claims to interpret itself, and the same one chapter 03 raised as a Gettier problem for sensors and pipelines. What Curry–Howard delivers instead is certainty re-engineered as a budget: instead of trusting a 250-page argument and twelve tired referees, trust a kernel of a few thousand lines, state exactly what would have to fail for the proof to be wrong, and let that be the whole of the remaining doubt.

Philosophy wanted proof to end an argument once and for all; computation's counteroffer is a proof that tells you precisely how small the argument left over actually is.

> ไม่มีสิ่งใดในนี้ส่งมอบความแน่นอนที่ Hilbert ต้องการในปี 1930 ได้ การรับประกันของ proof assistant แข็งแรงได้แค่เท่า kernel ของมัน และ kernel ก็วางอยู่บนคอมไพเลอร์ ซึ่งวางอยู่บนฮาร์ดแวร์ ซึ่งพังได้ — การถอยกลับไม่รู้จบ (regress) แบบเดียวกับที่ Wittgenstein กดดันกฎใดๆ ที่อ้างว่าตีความตัวเองได้ และแบบเดียวกับที่บทที่ 03 ยกขึ้นมาเป็นปัญหาเก็ตติเยร์ (Gettier problem) สำหรับเซนเซอร์และไปป์ไลน์ สิ่งที่ Curry–Howard ส่งมอบให้แทนคือความแน่นอนที่ถูกออกแบบใหม่ให้เป็นงบประมาณ: แทนที่จะไว้ใจข้อโต้แย้ง 250 หน้ากับผู้ตรวจทานที่เหนื่อยล้าสิบสองคน ก็ไว้ใจ kernel ไม่กี่พันบรรทัด ระบุให้ชัดว่าอะไรต้องพังถึงจะทำให้บทพิสูจน์ผิด แล้วปล่อยให้นั่นเป็นความสงสัยที่เหลืออยู่ทั้งหมด ปรัชญาต้องการให้บทพิสูจน์ยุติข้อโต้แย้งได้เด็ดขาดครั้งเดียวจบ ข้อเสนอโต้กลับของการคำนวณคือบทพิสูจน์ที่บอกได้แม่นยำว่าข้อโต้แย้งที่เหลืออยู่นั้นเล็กแค่ไหนจริงๆ

Connections · เชื่อมโยง

Chapter 01 introduces Curry–Howard as the logic chapter's treaty between the two disciplines; this chapter completes it. Chapter 03 is where the four-color theorem first caused its epistemological shock — this chapter is its answer. Chapter 09 supplies the incompleteness results that proof assistants must live inside rather than escape. Chapter 11's Kolmogorov complexity measures the length of a proof the way this chapter measures its trustworthiness.

> บทที่ 01 แนะนำความสอดคล้อง Curry–Howard (Curry–Howard correspondence) ในฐานะสนธิสัญญาของบทตรรกศาสตร์ระหว่างสองสาขาวิชา บทนี้ทำให้มันสมบูรณ์ บทที่ 03 คือจุดที่ทฤษฎีบทสี่สีก่อแรงสั่นสะเทือนเชิงญาณวิทยาเป็นครั้งแรก — บทนี้คือคำตอบของมัน บทที่ 09 ให้ผลลัพธ์เรื่องความไม่สมบูรณ์ที่ proof assistant ต้องอยู่ข้างในแทนที่จะหนีออกไปได้ Kolmogorov complexity ในบทที่ 11 วัดความยาวของบทพิสูจน์ ในแบบเดียวกับที่บทนี้วัดความน่าเชื่อถือของมัน

Open questions · คำถามที่ยังเปิดอยู่

If a kernel can be made small enough for one person to audit, has the regress of trusting the verifier actually been stopped, or only shortened to a link short enough to overlook? Does univalence settle what "the same structure" means in mathematics generally, or only inside type theory's own house? And now that Lean's mathlib checks new results as they are proved, does mathematical knowledge still need the human community of referees at all, or only the community that decides which theorems are worth proving in the first place?

> ถ้า kernel ทำให้เล็กพอสำหรับคนคนเดียวตรวจสอบได้ การถอยกลับไม่รู้จบของการไว้ใจตัวตรวจสอบถูกหยุดจริงๆ แล้วหรือยัง หรือแค่ถูกย่อให้สั้นลงจนเหลือจุดเชื่อมที่สั้นพอจะมองข้ามไปเฉยๆ? สัจพจน์เอกค่า (univalence) ยุติความหมายของ "โครงสร้างเดียวกัน" ในคณิตศาสตร์โดยทั่วไปได้จริงหรือ หรือยุติได้แค่ภายในบ้านของทฤษฎีชนิดเองเท่านั้น? และตอนนี้ที่ mathlib ของ Lean ตรวจสอบผลลัพธ์ใหม่ไปพร้อมกับที่มันถูกพิสูจน์ ความรู้ทางคณิตศาสตร์ยังต้องการชุมชนมนุษย์ของผู้ตรวจทานอยู่ไหม หรือต้องการแค่ชุมชนที่ตัดสินว่าทฤษฎีบทไหนควรค่าแก่การพิสูจน์ตั้งแต่แรกเท่านั้น?

Go deeper · อ่านต่อ

— Jean-Yves Girard, Yves Lafont, and Paul Taylor, *Proofs and Types*, Cambridge Tracts in Theoretical Computer Science 7 (Cambridge: Cambridge University Press, 1989) — free edition at paultaylor.eu/stable/prot.pdf.

> Jean-Yves Girard, Yves Lafont, and Paul Taylor, Proofs and Types, Cambridge Tracts in Theoretical Computer Science 7 (Cambridge: Cambridge University Press, 1989) — free edition at paultaylor.eu/stable/prot.pdf.

— William Howard, "The Formulae-as-Types Notion of Construction," in *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*, ed. J. P. Seldin and J. R. Hindley (New York: Academic Press, 1980), 479–490; Georges Gonthier, "Formal Proof — The Four-Color Theorem," *Notices of the American Mathematical Society* 55 (2008): 1382–1393 — ams.org/notices/200811/tx081101382p.pdf.

> William Howard, "The Formulae-as-Types Notion of Construction," in To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism, ed. J. P. Seldin and J. R. Hindley (New York: Academic Press, 1980), 479–490; Georges Gonthier, "Formal Proof — The Four-Color Theorem," Notices of the American Mathematical Society 55 (2008): 1382–1393 — ams.org/notices/200811/tx081101382p.pdf.

— Stanford Encyclopedia of Philosophy, "Intuitionism in the Philosophy of Mathematics" — plato.stanford.edu/entries/intuitionism.

> Stanford Encyclopedia of Philosophy, "Intuitionism in the Philosophy of Mathematics" — plato.stanford.edu/entries/intuitionism.

CHAPTER 13

Learning Machines

เครื่องเรียนรู้

This is Part II's fifth hearing: computation has spent a decade teaching itself to speak, translate, and see without being told a single rule, and now philosophy is asked to explain how that could count as knowing anything.

นี่คือการไต่สวนครั้งที่ 5 ของภาค II: การคำนวณใช้เวลาสิบปีสอนตัวเองให้พูด แปล และมองเห็น โดยไม่มีใครบอกกฎสักข้อเดียว และตอนนี้ปรัชญาถูกขอให้อธิบายว่าสิ่งนั้นจะนับเป็นการรู้อะไรบางอย่างได้อย่างไร

What does machine learning prove about how knowledge enters a mind?

การเรียนรู้ของเครื่องพิสูจน์อะไรได้บ้างเกี่ยวกับวิธีที่ความรู้เข้าสู่จิต

On 24 May 2024, Anthropic let the public talk to a model that could not stop mentioning the Golden Gate Bridge. Ask it for a love letter and it worked the bridge's orange paint into a metaphor; ask it to file your taxes and it found a way back to the fog over the strait. This was "Golden Gate Claude," a demonstration built from research the company had just published: "Scaling Monosemanticity," which used a sparse-dictionary technique to pull tens of millions of human-interpretable "features" out of the middle layers of Claude 3 Sonnet — internal directions that reliably fire for a specific bridge, for sycophancy, for a known kind of security flaw (Anthropic, "Scaling Monosemanticity," May 2024). For the demonstration, one feature was simply turned up until it dominated. The model was taken offline about a day later. What it left behind was stranger than the joke: a single concept, inside a system nobody had hand-coded concept by concept, had an ad-

dress you could dial like a volume knob. A structure philosophers had called irretrievably opaque turned out, at least at this one coordinate, to have a location.

> วันที่ 24 พฤษภาคม 2024 Anthropic เปิดให้สาธารณะพูดคุยกับโมเดลที่พูดถึง Golden Gate Bridge อยู่ตลอดจนหยุดไม่ได้ ขอให้มันเขียนจดหมายรัก มันก็ยังดึงสีส้มของสะพานมาใช้เป็นอุปมาได้ ขอให้มันช่วยยื่นภาษี มันก็ยังหาทางวกกลับไปที่หมอกเหนือช่องแคบจนได้ นี่คือ "Golden Gate Claude" การสาธิตที่สร้างจากงานวิจัยที่บริษัทเพิ่งเผยแพร่ไป: "Scaling Monosemanticity" ซึ่งใช้เทคนิค sparse-dictionary ดึง "feature" ที่มนุษย์ตีความได้นับสิบล้านตัวออกมาจากเลเยอร์กลางของ Claude 3 Sonnet — ทิศทางภายในที่ทำงานสม่ำเสมอเมื่อพบสะพานเส้นนั้นเส้นเดียว เมื่อพบความประจบสอพลอ หรือเมื่อพบช่องโหว่ความปลอดภัยแบบที่รู้จักกันอยู่แล้ว (Anthropic, "Scaling Monosemanticity," May 2024) สำหรับการสาธิตครั้งนี้ feature ตัวหนึ่งถูกเร่งขึ้นเรื่อยๆ จนครอบงำคำตอบทั้งหมด โมเดลตัวนี้ถูกถอดออกจากระบบภายในเวลาประมาณหนึ่งวันถัดมา สิ่งที่มันทิ้งไว้แปลกกว่ามุกตลกนั้นเสียอีก: แนวคิดเดียวภายในระบบที่ไม่มีใครเขียนโค้ดทีละแนวคิดด้วยมือ กลับมีที่อยู่ให้หมุนปรับได้เหมือนปุ่มเสียง โครงสร้างที่นักปรัชญาเคยเรียกว่าทึบจนกู้คืนไม่ได้กลับกลายเป็นว่า อย่างน้อยที่พิกัดจุดนี้จุดเดียว มันมีตำแหน่งที่ตั้งอยู่จริง

That location is where this chapter starts, because finding it reopens a dispute older than any computer: what, if anything, must a mind already have before experience can teach it something.

> ตำแหน่งนั้นคือจุดที่บทนี้เริ่มต้น เพราะการพบมันเปิดข้อพิพาทเก่าแก่กว่าคอมพิวเตอร์เครื่องไหนขึ้นมาใหม่: จิตต้องมีอะไรติดตัวมาก่อน ถ้ามี ก่อนที่ประสบการณ์จะสอนอะไรมันได้

Empiricism and rationalism have been run, this past decade, as an experiment at scale. Noam Chomsky coined "poverty of the stimulus" in *Rules and Representations* (1980): children acquire grammatical structure that the input they actually hear radically underdetermines, so some of that structure must be innate — Universal Grammar. Large language models seem to answer back: trained on nothing but text, with no grammar built in, they be-

come fluent. But the reply does not settle the argument, and philosophy objects on the argument's own terms. Chomsky, with Berwick, Pietroski, and Yankama, restated the case in "Poverty of the Stimulus Revisited" (2011): the claim was always about what a *child*'s actual, limited exposure can support, not about what any learner given enough data can do. A model trained on a written record no child ever sees is not a counterexample to a claim about children; it is a different experiment with a different subject.

> ประสบการณ์นิยมกับเหตุผลนิยมถูกนำมาทดลองในขนาดใหญ่ตลอดสิบปีที่ผ่านมา Noam Chomsky บัญญัติคำว่า "ความขาดแคลนของสิ่งเร้า" (poverty of the stimulus) ไว้ใน Rules and Representations (1980): เด็กเรียนรู้โครงสร้างไวยากรณ์ที่ข้อมูลนำเข้าซึ่งพวกเขาได้ยินจริงกำหนดไว้ไม่พอเลย ฉะนั้นโครงสร้างบางส่วนต้องติดตัวมาแต่กำเนิด — ไวยากรณ์สากล (Universal Grammar) โมเดลภาษาขนาดใหญ่ดูเหมือนจะตอบโต้กลับ: เทรนด้วยข้อความล้วนๆ ไม่มีไวยากรณ์ฝังไว้ในตัวเลย ก็ยังพูดได้คล่องแคล่ว แต่คำตอบนี้ไม่ได้ปิดข้อถกเถียง และปรัชญาก็คัดค้านด้วยเงื่อนไขของข้อโต้แย้งเอง Chomsky ร่วมกับ Berwick, Pietroski และ Yankama กล่าวข้อเสนอนี้ซ้ำใน "Poverty of the Stimulus Revisited" (2011): ข้อกล่าวอ้างนี้พูดถึงสิ่งที่การรับข้อมูลจริงและจำกัดของ เด็ก รองรับได้เสมอมา ไม่ใช่สิ่งที่ผู้เรียนรู้คนไหนก็ตามทำได้ถ้าได้ข้อมูลมากพอ โมเดลที่เทรนจากบันทึกลายลักษณ์อักษรซึ่งไม่มีเด็กคนไหนเคยเห็น ไม่ใช่ตัวอย่างค้านข้อกล่าวอ้างที่พูดถึงเด็ก แต่มันคือการทดลองคนละอันกับหัวข้อคนละเรื่อง

The empiricist side of that experiment has a doctrine, and the doctrine is a wager about how knowledge arrives. Richard Sutton put it into engineering form in March 2019: "The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin" (Sutton, "The Bitter Lesson," 13 March 2019). Chess, Go, speech, and vision each fell to search and learning after decades spent hand-encoding what researchers thought minds needed. But a lesson drawn from which strategies won in a handful of do-

mains is a historical generalization, not a proof about how any mind must gain knowledge — and it is silent on domains, like formal proof (ch.12), where built-in structure has not obviously become a liability.

> ฝั่งประสบการณ์นิยมของการทดลองนี้มีหลักคำสอนของตัวเอง และหลักคำสอนนั้นคือการพนันว่าความรู้มาถึงได้อย่างไร Richard Sutton แปลงมันเป็นรูปแบบเชิงวิศวกรรมในเดือนมีนาคม 2019: "บทเรียนที่ใหญ่ที่สุดที่อ่านได้จากงานวิจัย AI 70 ปี คือวิธีการทั่วไปที่ใช้ประโยชน์จาก compute ได้ผลดีที่สุดในที่สุด และดีกว่าด้วยระยะห่างมาก" ("The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin") (Sutton, "The Bitter Lesson," 13 March 2019) หมากรุกสากล โกะ การพูด และการมองเห็น ต่างพ่ายให้กับการค้นหาและการเรียนรู้ หลังจากหลายทศวรรษที่ใช้ไปกับการเข้ารหัสด้วยมือในสิ่งที่นักวิจัยคิดว่าจิตต้องการ แต่บทเรียนที่สรุปมาจากว่ากลยุทธ์ไหนชนะในโดเมนจำนวนหนึ่งเป็นเพียงการสรุปเชิงประวัติศาสตร์ ไม่ใช่บทพิสูจน์ว่าจิตใดๆ ต้องได้ความรู้มาอย่างไร — และมันก็ไม่พูดถึงโดเมนอย่างบทพิสูจน์เชิงรูปนัย (บทที่ 12) ที่โครงสร้างซึ่งฝังมาแต่ต้นยังไม่กลายเป็นภาระอย่างชัดเจน

Chapter 05 puts Stevan Harnad's 1990 question on the record: can a system that manipulates symbols purely by their shape ever mean anything by them, or does it only borrow meaning from whoever reads its output? A model trained only on text is, on that question, purer than anything Harnad had in mind: its entire "experience" is other symbols. Is a feature that fires reliably on "Golden Gate Bridge" across languages and phrasings a grounded reference to the object, or an extremely well-organized cluster of shapes that still refers to nothing? The finding is evidence; it is not a verdict.

> บทที่ 05 บันทึกคำถามของ Stevan Harnad ในปี 1990 ไว้เป็นหลักฐาน: ระบบที่จัดการสัญลักษณ์ด้วยรูปทรงล้วนๆ จะมีความหมายอะไรในตัวมันได้จริงหรือไม่ หรือมันแค่ยืมความหมายมาจากใครก็ตามที่อ่านผลลัพธ์ของมัน โมเดลที่เทรนจากข้อความล้วนๆ ตอบคำถามนี้ได้บริสุทธิ์ยิ่งกว่าอะไรที่ Harnad เคยนึกถึง เพราะ "ประสบการณ์" ทั้งหมดของมันคือสัญลักษณ์อื่นๆ feature ที่ทำงานสม่ำเสมอทุกครั้ง

> ที่พบ "Golden Gate Bridge" ข้ามภาษาและข้ามสำนวน คือการอ้างอิงที่ยึดโยงกับวัตถุจริง หรือเป็นแค่กลุ่มก้อนรูปทรงที่จัดระเบียบมาอย่างดีมาก แต่ก็ยังไม่ได้อ้างอิงถึงอะไรเลย ข้อค้นพบนี้เป็นหลักฐาน ไม่ใช่คำตัดสิน

The research line runs from "A Mathematical Framework for Transformer Circuits" (Anthropic, December 2021) through "Towards Monosemanticity" (2023) to the 2024 scaling result, and it gives philosophy of mind something it rarely has: a system whose internal representations were *discovered*, not designed, open to direct inspection rather than armchair argument. The caution belongs beside the excitement — a "feature" is what a sparse autoencoder, itself a trained model, finds; whether that unit deserves the name "concept" is exactly the question interpretability opens rather than closes.

> สายงานวิจัยนี้ไล่มาตั้งแต่ "A Mathematical Framework for Transformer Circuits" (Anthropic, ธันวาคม 2021) ผ่าน "Towards Monosemanticity" (2023) มาจนถึงผลลัพธ์การขยายสเกลในปี 2024 และมันมอบสิ่งที่ปรัชญาแห่งจิตแทบไม่เคยมี: ระบบที่การแทนค่าภายในถูก ค้นพบ ไม่ใช่ถูกออกแบบ เปิดให้ตรวจสอบได้โดยตรงแทนที่จะเป็นแค่การถกเถียงบนเก้าอี้ ความระมัดระวังต้องมาคู่กับความตื่นเต้น — "feature" คือสิ่งที่ sparse autoencoder ซึ่งตัวมันเองก็เป็นโมเดลที่ผ่านการเทรนมา ค้นพบ ส่วนหน่วยนั้นสมควรได้ชื่อว่า "concept" หรือไม่ คือคำถามที่ interpretability (การตีความได้) เปิดไว้ ไม่ใช่ปิดลง

Hume's problem is replayed, batch by batch, inside the training loop. In 1748, Hume showed that no argument — demonstrative or probable — can license projecting a past regularity forward without already assuming nature is uniform, which is the very thing in question (Hume, *An Enquiry Concerning Human Understanding*, Section IV, 1748). Every gradient update is exactly that projection: a rule fit to past batches, licensed to govern data it has never seen. Statistical learning theory supplies generalization bounds, but only by assuming train and test are drawn from the same distribution — Hume's uniformity principle, formalized and *assumed*, not derived. Computing's retort is that probably-approximately-correct is enough to ship; philo-

sophy's is that this is enough only because the premise Hume showed unprovable was smuggled in at the start, which is exactly why models fail silently the moment the world stops resembling their training data.

> ปัญหาของ Hume ถูกเล่นซ้ำทีละแบตช์อยู่ภายใน training loop ในปี 1748 Hume แสดงให้เห็นว่าไม่มีข้อโต้แย้งใด — ไม่ว่าจะพิสูจน์ได้แน่นอนหรือแค่น่าจะเป็นไปได้ — ที่จะอนุญาตให้ฉายภาพความสม่ำเสมอในอดีตไปข้างหน้าได้ โดยไม่สมมติไว้ก่อนแล้วว่าธรรมชาติมีความสม่ำเสมอ ซึ่งนั่นคือสิ่งที่กำลังถูกตั้งคำถามอยู่พอดี (Hume, An Enquiry Concerning Human Understanding, Section IV, 1748) การอัปเดต gradient ทุกครั้งคือการฉายภาพแบบนั้นเป๊ะๆ: กฎที่ fit กับแบตช์ในอดีต แล้วได้รับอนุญาตให้ควบคุมข้อมูลที่มันไม่เคยเห็นมาก่อน ทฤษฎีการเรียนรู้เชิงสถิติให้ขอบเขตการทั่วไป (generalization bound) มาได้ก็จริง แต่ต้องสมมติว่าชุด train กับชุด test ถูกดึงมาจากการกระจายตัวเดียวกันเท่านั้น — หลักความสม่ำเสมอของ Hume ที่ถูกทำให้เป็นรูปนัยและ สมมติไว้ ไม่ใช่พิสูจน์ได้ คำโต้ของฝั่งการคำนวณคือแค่ probably-approximately-correct ก็พอจะส่งของได้แล้ว ส่วนคำโต้ของปรัชญาคือมันพอเพราะข้อสมมติที่ Hume แสดงว่าพิสูจน์ไม่ได้ถูกลักลอบใส่เข้ามาตั้งแต่ต้นเท่านั้น ซึ่งนั่นคือเหตุผลตรงๆ ว่าทำไมโมเดลถึงล้มเหลวแบบเงียบๆ ทันทีที่โลกเลิกคล้ายกับข้อมูลที่มันเทรนมา

The people, briefly. Sutton has spent four decades building reinforcement learning's core toolkit — temporal-difference learning, actor-critic methods — from the University of Massachusetts to AT&T Labs to the University of Alberta, where he has been a professor since 2003 and, from 2017, a distinguished scientist at DeepMind Alberta. Chomsky has argued the rationalist case from MIT linguistics since the 1950s. Harnad's grounding paper has outlived several fashions in AI it was written to challenge. Chris Olah, an Anthropic co-founder who previously worked on interpretability at Google Brain and OpenAI, has run the Transformer Circuits project since its 2021 opening paper.

> **ผู้คน โดยย่อ** Sutton ใช้เวลาสี่ทศวรรษสร้างชุดเครื่องมือหลักของ reinforcement learning — temporal-difference learning, actor-critic methods — ตั้งแต่ University of Massachusetts ไป AT&T Labs จนถึง University of Alberta ที่เขา

เป็นศาสตราจารย์มาตั้งแต่ปี 2003 และตั้งแต่ปี 2017 เป็นนักวิทยาศาสตร์ดีเด่นที่ DeepMind Alberta Chomsky ยืนกรานฝ่ายเหตุผลนิยมจากภาควิชาภาษาศาสตร์ที่ MIT มาตั้งแต่ทศวรรษ 1950 เปเปอร์เรื่องการยึดโยงของ Harnad อยู่ยงคงกระพันมากว่ากระแสหลายกระแสใน AI ที่มันถูกเขียนขึ้นมาท้าทาย Chris Olah ผู้ร่วมก่อตั้ง Anthropic ที่เคยทำงานด้าน interpretability ที่ Google Brain และ OpenAI มาก่อน เป็นผู้ดูแลโครงการ Transformer Circuits มาตั้งแต่เปเปอร์เปิดตัวในปี 2021

Connections · เชื่อมโยง

Chapter 04 asked whether the mind is what the brain computes; this chapter asks what an artificial learner's success or failure can actually prove about that question, and mostly finds "less than advertised." Chapter 05 owns the symbol grounding problem this chapter tests at scale. Chapter 09 supplies the notion of mechanical procedure this whole debate assumes. Chapter 10 supplies the bounded agent that Sutton's engineering doctrine implicitly assumes and classical epistemology did not. Chapter 11 explains the bracket Shannon drew around meaning; interpretability is computing's first serious attempt to look back inside it.

บทที่ 04 ถามว่าจิตคือสิ่งที่สมองคำนวณหรือไม่ บทนี้ถามว่าความสำเร็จหรือความล้มเหลวของผู้เรียนรู้เทียมพิสูจน์อะไรได้จริงเกี่ยวกับคำถามนั้น และส่วนใหญ่พบว่า "น้อยกว่าที่โฆษณาไว้" บทที่ 05 เป็นเจ้าของปัญหาการยึดโยงสัญลักษณ์ (symbol grounding problem) ที่บทนี้นำมาทดสอบในขนาดใหญ่ บทที่ 09 ให้แนวคิดเรื่องกระบวนวิธีเชิงกลไกที่ข้อถกเถียงทั้งหมดนี้สมมติไว้เป็นฐาน บทที่ 10 ให้ตัวแทนที่มีขอบเขต (bounded agent) ซึ่งหลักคำสอนเชิงวิศวกรรมของ Sutton สมมติไว้โดยปริยาย แต่ญาณวิทยาแบบคลาสสิกไม่เคยสมมติ บทที่ 11 อธิบายวงเล็บที่ Shannon ขีดไว้รอบความหมาย interpretability คือความพยายามจริงจังครั้งแรกของฝั่งการคำนวณที่จะมองย้อนกลับเข้าไปข้างในวงเล็บนั้น

Open questions · คำถามที่ยังเปิดอยู่

Would any degree of language-model fluency count as evidence against poverty-of-the-stimulus, or is the argument, properly read, immune to that kind of evidence by construction? Does a feature that survives translation and paraphrase amount to reference, or only to a very tight correlation mistaken for one? If the bitter lesson is right that hand-built priors lose out in the long run, what should we make of the domains — so far — where they have not?

> ความคล่องแคล่วของโมเดลภาษาระดับไหนก็ตามจะนับเป็นหลักฐานค้าน poverty of the stimulus ได้หรือไม่ หรือว่าข้อโต้แย้งนี้ ถ้าอ่านให้ถูกต้อง มีภูมิคุ้มกันต่อหลักฐานแบบนั้นอยู่ในตัวโครงสร้างของมันเองอยู่แล้ว feature ที่รอดจากการแปลและการเปลี่ยนสำนวนนับเป็นการอ้างอิงได้จริงหรือไม่ หรือเป็นแค่ความสัมพันธ์ที่แนบแน่นมากจนถูกเข้าใจผิดว่าเป็นการอ้างอิง ถ้า the bitter lesson ถูกต้องที่ว่า prior ที่สร้างด้วยมือแพ้ในระยะยาว แล้วเราจะว่าอย่างไรกับโดเมนต่างๆ ที่ — จนถึงตอนนี้ — มันยังไม่แพ้

Go deeper · อ่านต่อ

— Noam Chomsky, *Rules and Representations*, Columbia University Press, 1980.

> Noam Chomsky, Rules and Representations, Columbia University Press, 1980.

— Richard Sutton, "The Bitter Lesson," 13 March 2019 — incompleteideas.net/IncIdeas/BitterLesson.html; Anthropic, "Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet," May 2024 — transformer-circuits.pub/2024/scaling-monosemanticity.

> Richard Sutton, "The Bitter Lesson," 13 March 2019 — incompleteideas.net/IncIdeas/BitterLesson.html; Anthropic, "Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet," May 2024 — transformer-circuits.pub/2024/scaling-monosemanticity.

— *Rationalism vs. Empiricism*, Stanford Encyclopedia of Philosophy.

> Rationalism vs. Empiricism, Stanford Encyclopedia of Philosophy.

CHAPTER 14

Distributed Trust

ความไว้วางใจแบบกระจาย

This is Part II's sixth hearing: computation has spent forty years proving, formally, exactly how much agreement strangers with no shared authority can and cannot reach — and philosophy's social contract is called to answer for what it left out.

นี่คือการไต่สวนครั้งที่ 6 ของภาค II: การคำนวณใช้เวลาสี่สิบปีพิสูจน์อย่างเป็นรูปนัยว่าคนแปลกหน้าที่ไม่มีอำนาจร่วมกันจะตกลงกันได้มากแค่ไหนและตกลงกันไม่ได้แค่ไหน — และสัญญาประชาคมของปรัชญาถูกเรียกมาตอบคำถามว่ามันละเลยอะไรไป

Can computers settle what philosophers only argued — how strangers cooperate?

คอมพิวเตอร์ยุติสิ่งที่นักปรัชญาเคยแค่ถกเถียงกันได้หรือไม่ — เรื่องที่ว่าคนแปลกหน้าร่วมมือกันได้อย่างไร

On 3 January 2009, a program mined the first block of a currency that no bank, state, or company controlled. Embedded in that block was a string of text: "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks" — the front-page headline of that day's London *Times*, quoting the news that the UK government was about to bail out its banking system a second time (Bitcoin Wiki, "Genesis block"). The line proved the block could not have been mined earlier, but it did more than timestamp; it dated the system's birth to the exact week the institutions trust is normally outsourced to had just failed in public. The paper behind it, "Bitcoin: A Peer-to-Peer Electronic Cash System," had been posted to a cryptography mailing list on 31 October 2008 under the name Satoshi Nakamoto — a name nobody has ever conclusively attached to a person. A system built explicitly

so that strangers need not trust one another was written by someone none of us can authenticate by any of the ordinary means. That irony is this chapter's opening exhibit.

> วันที่ 3 มกราคม 2009 โปรแกรมหนึ่งขุดบล็อกแรกของสกุลเงินที่ไม่มีธนาคาร รัฐ หรือบริษัทใดควบคุมได้ ฝังอยู่ในบล็อกนั้นคือข้อความ: "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks" — พาดหัวหน้าหนึ่งของหนังสือพิมพ์ The Times กรุงลอนดอนในวันนั้น อ้างข่าวว่ารัฐบาลสหราชอาณาจักรกำลังจะอุ้มระบบธนาคารของตัวเองเป็นครั้งที่สอง (Bitcoin Wiki, "Genesis block") ข้อความบรรทัดนี้พิสูจน์ว่าบล็อกนี้ไม่มีทางถูกขุดก่อนหน้านั้นได้ แต่มันทำมากกว่าแค่ประทับเวลา มันระบุวันเกิดของระบบไว้ตรงกับสัปดาห์ที่สถาบันซึ่งความไว้วางใจมักถูกฝากไว้ด้วยเพิ่งล้มเหลวต่อหน้าสาธารณะพอดี เปเปอร์เบื้องหลังเรื่องนี้ "Bitcoin: A Peer-to-Peer Electronic Cash System" ถูกโพสต์ลงเมลลิ่งลิสต์ด้าน cryptography เมื่อวันที่ 31 ตุลาคม 2008 ภายใต้ชื่อ Satoshi Nakamoto — ชื่อที่ไม่มีใครเคยเชื่อมโยงกับตัวบุคคลได้อย่างชี้ขาด ระบบที่ถูกสร้างขึ้นมาโดยตรงเพื่อให้คนแปลกหน้าไม่ต้องไว้ใจกัน กลับถูกเขียนขึ้นโดยใครสักคนที่ไม่มีใครในพวกเราสามารถยืนยันตัวตนได้ด้วยวิธีปกติใดๆ เลย ความย้อนแย้งนั้นคือหลักฐานชิ้นแรกที่บทนี้นำเข้าสู่การพิจารณา

The social contract has an executable version, and it begins from the same premise the original did. Thomas Hobbes argued in *Leviathan* (1651) that without an authority capable of enforcing it, agreement among equals is unstable — the state of nature is "a war of all against all," and peace requires transferring judgment to a sovereign strong enough to compel it. Distributed computing's answer, worked out three centuries later, keeps the peace without a sovereign: Leslie Lamport's Paxos ("The Part-Time Parliament," *ACM Transactions on Computer Systems*, May 1998) and Diego Ongaro and John Ousterhout's Raft ("In Search of an Understandable Consensus Algorithm," USENIX ATC 2014) let any majority of equal replicas bind a decision, tolerating the minority being slow, wrong, or offline. Where Hobbes

solved coordination by concentrating judgment in one place, these protocols solve it by counting votes among peers — a social contract that needs a quorum and a clock, never a Leviathan.

> สัญญาประชาคมมีเวอร์ชันที่รันได้จริง และมันเริ่มจากข้อสมมติเดียวกับต้นฉบับ Thomas Hobbes โต้แย้งใน Leviathan (1651) ว่าถ้าไม่มีอำนาจที่บังคับใช้ได้จริง ข้อตกลงระหว่างผู้เท่าเทียมกันย่อมไม่มั่นคง — สภาวะธรรมชาติคือ "สงครามของทุกคนต่อทุกคน" และสันติภาพต้องอาศัยการโอนอำนาจตัดสินใจไปให้ผู้มีอธิปไตยที่แข็งแกร่งพอจะบังคับมันได้ คำตอบของ distributed computing ที่คิดค้นขึ้นสามศตวรรษต่อมา รักษาสันติภาพไว้ได้โดยไม่ต้องมีผู้มีอธิปไตยเลย: Paxos ของ Leslie Lamport ("The Part-Time Parliament," ACM Transactions on Computer Systems, May 1998) และ Raft ของ Diego Ongaro กับ John Ousterhout ("In Search of an Understandable Consensus Algorithm," USENIX ATC 2014) ให้เสียงข้างมากของ replica ที่เท่าเทียมกันผูกมัดการตัดสินใจได้ ทนต่อการที่เสียงข้างน้อยจะช้า ผิดพลาด หรือออฟไลน์ไป จุดที่ Hobbes แก้ปัญหาการประสานงานด้วยการรวมอำนาจตัดสินใจไว้ที่จุดเดียว โปรโตคอลเหล่านี้แก้ด้วยการนับคะแนนเสียงในหมู่ผู้เท่าเทียมกัน — สัญญาประชาคมที่ต้องการแค่ quorum กับนาฬิกา ไม่ต้องการ Leviathan เลย

Testimony gets harder when a witness may be lying rather than merely mistaken. Leslie Lamport, Robert Shostak, and Marshall Pease posed that sharper version in 1982: generals surrounding a city must reach unanimous agreement — attack or retreat — communicating only by messenger, when some generals or their messages may be actively treacherous. Their result: with unsigned, oral messages, no protocol works unless more than two-thirds of the generals are loyal; among just three participants, a single traitor already defeats two honest ones (Lamport, Shostak & Pease, "The Byzantine Generals Problem," *ACM TOPLAS* 4, no. 3, July 1982: 382–401).

Philosophy's testimony debates — how much a report should be discounted, knowing some sources may be adversarial rather than merely unreliable — get an exact numerical threshold: one-third.

> คำให้การยิ่งยากขึ้นเมื่อพยานอาจกำลังโกหก ไม่ใช่แค่เข้าใจผิด Leslie Lamport, Robert Shostak และ Marshall Pease ตั้งเวอร์ชันที่คมกว่านั้นไว้ในปี 1982: นายพลที่ล้อมเมืองไว้ต้องบรรลุข้อตกลงเป็นเอกฉันท์ — บุกหรือถอย — โดยสื่อสารกันผ่านผู้ส่งสารเท่านั้น ในสถานการณ์ที่นายพลบางคนหรือข้อความบางฉบับอาจเป็นการทรยศจริงๆ ผลลัพธ์ของพวกเขาคือ: ด้วยข้อความปากเปล่าที่ไม่มีลายเซ็นกำกับ จะไม่มีโปรโตคอลไหนใช้ได้เลย เว้นแต่นายพลมากกว่าสองในสามจะซื่อสัตย์ ในกลุ่มแค่สามคน ผู้ทรยศเพียงคนเดียวก็เอาชนะผู้ซื่อสัตย์สองคนได้แล้ว (Lamport, Shostak & Pease, "The Byzantine Generals Problem," ACM TOPLAS 4, no. 3, July 1982: 382–401) ข้อถกเถียงเรื่องคำให้การของปรัชญา — ควรลดน้ำหนักรายงานลงแค่ไหน เมื่อรู้ว่าบางแหล่งอาจเป็นปรปักษ์ ไม่ใช่แค่ไม่น่าเชื่อถือ — ได้เกณฑ์ตัวเลขที่ชัดเจน: หนึ่งในสาม

Agreeing on a plan is one thing; agreeing that you have both agreed is another, and it is provably out of reach. In 1975, Akkoyunlu, Ekanadham, and Huber proved that two parties coordinating over a channel that can silently drop messages can never become certain, after any finite exchange, that both sides know the plan is on — any acknowledgment could itself be the message that is lost. Jim Gray gave the result its familiar military name, the Two Generals Problem, in 1978. This is the philosophical structure of common knowledge — I know P, you know P, I know you know P, without end — which David Lewis had formalized in *Convention* (1969); Joseph Halpern and Yoram Moses gave it a distributed-computing treatment, proving that true common knowledge is unreachable over any channel with even a small chance of failure, and that only finite, nested approximations are ("Know-

ledge and Common Knowledge in a Distributed Environment," 1984; expanded *Journal of the* ACM 37, no. 3, 1990). Epistemic logic, usually a philosopher's diagram of nested belief, becomes a protocol with a proven ceiling.

> การตกลงกันเรื่องแผนคือเรื่องหนึ่ง การตกลงกันว่าทั้งสองฝ่ายตกลงกันแล้วคืออีกเรื่องหนึ่ง และเรื่องหลังนี้พิสูจน์ได้ว่าไปไม่ถึงเลย ในปี 1975 Akkoyunlu, Ekanadham และ Huber พิสูจน์ว่าสองฝ่ายที่ประสานงานผ่านช่องทางที่อาจทำข้อความหายไปเงียบๆ ได้ ไม่มีทางแน่ใจได้เลยหลังการแลกเปลี่ยนจำนวนจำกัดใดๆ ว่าทั้งสองฝ่ายรู้ว่าแผนเดินหน้าแล้ว เพราะการตอบรับใดๆ ก็อาจเป็นข้อความที่หายไปเสียเอง Jim Gray ตั้งชื่อทางการทหารที่คุ้นหูให้ผลลัพธ์นี้ว่า ปัญหานายพลสองนาย (Two Generals Problem) ในปี 1978 นี่คือโครงสร้างเชิงปรัชญาของความรู้ร่วม (common knowledge) — ฉันรู้ P อีกฝ่ายรู้ P ฉันรู้ว่าอีกฝ่ายรู้ P ไปเรื่อยไม่มีที่สิ้นสุด — ซึ่ง David Lewis ทำให้เป็นรูปนัยไว้ใน Convention (1969) Joseph Halpern และ Yoram Moses นำมันมาจัดการแบบ distributed computing พิสูจน์ว่าความรู้ร่วมที่แท้จริงไปไม่ถึงได้เลยผ่านช่องทางใดๆ ที่มีโอกาสล้มเหลวแม้เพียงเล็กน้อย และมีแต่การประมาณแบบซ้อนชั้นจำกัดเท่านั้นที่ไปถึงได้ ("Knowledge and Common Knowledge in a Distributed Environment," 1984; expanded Journal of the ACM 37, no. 3, 1990) ตรรกะเชิงญาณ (epistemic logic) ซึ่งปกติเป็นแค่แผนภาพความเชื่อซ้อนชั้นของนักปรัชญา กลายเป็นโปรโตคอลที่มีเพดานซึ่งพิสูจน์แล้ว

A digital signature lets any stranger verify who a message came from without a shared secret or a vouching intermediary: trust becomes a mathematical relation — only one private key produces what a matching public key can confirm — rather than a social one requiring a notary, a bank, or a state. A ledger extends this from verifying one signature to verifying an entire ordered history, letting mutually suspicious parties agree on a single sequence of events without deferring to any one of them.

> ลายเซ็นดิจิทัลทำให้คนแปลกหน้าคนไหนก็ตามยืนยันได้ว่าข้อความมาจากใคร โดยไม่ต้องมีความลับร่วมกันหรือตัวกลางผู้รับรอง: ความไว้วางใจกลายเป็นความสัมพันธ์เชิงคณิตศาสตร์ — มีเพียงกุญแจส่วนตัวดอกเดียวเท่านั้นที่สร้างสิ่งที่กุญแจสาธารณะคู่กันยืนยันได้ — แทนที่จะเป็นความสัมพันธ์เชิงสังคมที่ต้องมีทนายรับรอง ธนาคาร

> หรือรัฐ บัญชีแยกประเภท (ledger) ขยายเรื่องนี้จากการยืนยันลายเซ็นหนึ่งฉบับ ไปสู่การยืนยันประวัติทั้งชุดที่เรียงลำดับไว้ ทำให้ฝ่ายต่างๆ ที่ระแวงกันเองตกลงกันได้บนลำดับเหตุการณ์ชุดเดียว โดยไม่ต้องยอมตามฝ่ายใดฝ่ายหนึ่งเป็นพิเศษ

Money is where every piece of this machinery bears weight at once. Nakamoto's protocol makes the ledger itself the output of a Byzantine-fault-tolerant vote — proof-of-work — among anonymous, mutually distrustful participants, solving double-spending without the single trusted ledger-keeper Hobbes would have assumed was unavoidable. But philosophy objects here, not merely answers: the whole distributed-trust literature treats trust as solved once nodes agree on facts — which block is valid, whose signature is genuine. Older accounts of trust, including Hobbes's own worry about promises kept only under duress, hold that trust carries normative content no vote over present facts touches — reputation earned over years, forgiveness, the willingness to be vulnerable to someone else's future choice. A blockchain makes forgery costly. It does not make betrayal *within the rules* — a fraud run entirely inside a legitimate, signed ledger — visible to the protocol at all.

> เงินคือจุดที่ทุกชิ้นส่วนของกลไกนี้แบกน้ำหนักพร้อมกันทั้งหมด โปรโตคอลของ Nakamoto ทำให้ตัวบัญชีแยกประเภทเองเป็นผลผลิตของการโหวตแบบทนทานต่อความผิดพลาดแบบไบแซนไทน์ (Byzantine fault tolerance) — proof-of-work — ในหมู่ผู้เข้าร่วมนิรนามที่ไม่ไว้ใจกันเอง แก้ปัญหา double-spending ได้โดยไม่ต้องมีผู้ดูแลบัญชีที่น่าเชื่อถือเพียงรายเดียวแบบที่ Hobbes คงสมมติไว้ว่าเลี่ยงไม่ได้ แต่ปรัชญาคัดค้านตรงนี้ ไม่ใช่แค่ตอบคำถามเฉยๆ: วรรณกรรมเรื่อง distributed trust ทั้งหมดปฏิบัติต่อความไว้วางใจราวกับว่าแก้ไขได้แล้วทันทีที่โหนดตกลงกันเรื่องข้อเท็จจริง — บล็อกไหนถูกต้อง ลายเซ็นของใครเป็นของจริง ขณะที่คำอธิบายเรื่องความไว้วางใจแบบเก่ากว่า รวมถึงความกังวลของ Hobbes เองเรื่องคำสัญญาที่รักษาไว้ได้ก็เพราะถูกบังคับเท่านั้น ยืนยันว่าความไว้วางใจแบกเนื้อหาเชิงบรรทัดฐานที่การโหวตเรื่องข้อเท็จจริงปัจจุบันแตะไม่ถึงเลย — ชื่อเสียงที่สั่งสมมาหลายปี การให้อภัย ความเต็มใจที่จะเปราะบางต่อทางเลือกในอนาคตของอีกฝ่าย บล็อกเชนทำให้การ

> ปลอมแปลงมีต้นทุนสูง แต่มันไม่ได้ทำให้การทรยศ ภายในกรอบกฎ — การโกงที่เกิดขึ้นทั้งหมดภายในบัญชีแยกประเภทที่ถูกต้องตามกฎและมีลายเซ็นกำกับ — มองเห็นได้โดยโปรโตคอลเลยแม้แต่น้อย

The people, briefly. Hobbes wrote *Leviathan* out of the English Civil War's chaos. Lamport received the 2013 ACM Turing Award for this and other foundational work in distributed computing, and by his own account wrote "The Part-Time Parliament" as an allegory of a fictional Greek parliament because reviewers found the direct version unreadable. Halpern and Moses built on Lewis's philosophical account of convention to give epistemic logic a computational home. Nakamoto has never been identified.

> **ผู้คน โดยย่อ** Hobbes เขียน Leviathan ขึ้นจากความโกลาหลของสงครามกลางเมืองอังกฤษ Lamport ได้รับรางวัล ACM Turing Award ปี 2013 จากงานชิ้นนี้และงานพื้นฐานอื่นๆ ด้าน distributed computing และตามคำบอกเล่าของเขาเอง เขาเขียน "The Part-Time Parliament" เป็นนิทานเปรียบเทียบเกี่ยวกับรัฐสภากรีกสมมติ เพราะผู้ตรวจอ่านเห็นว่าเวอร์ชันตรงไปตรงมาอ่านไม่รู้เรื่อง Halpern และ Moses ต่อยอดจากคำอธิบายเชิงปรัชญาเรื่อง convention ของ Lewis เพื่อให้ตรรกะเชิงญาณมีบ้านทาง computational Nakamoto ไม่เคยถูกระบุตัวตนได้เลย

Connections · เชื่อมโยง

Chapter 03 asked whether a machine's answer counts as knowledge; Byzantine fault tolerance is that question with an exact fraction attached. Chapter 06's responsibility gaps recur here as blame distributed the same way trust is. Chapter 07's stack-versus-state tension sharpens once money itself no longer needs a state. Chapter 12's certainty-as-engineering-product reappears in the one-third threshold and the proven ceiling on common knowledge.

> บทที่ 03 ถามว่าคำตอบของเครื่องนับเป็นความรู้หรือไม่ ความทนทานต่อความผิดพลาดแบบไบแซนไทน์คือคำถามเดียวกันนั้นที่มีเศษส่วนตัวเลขแม่นยำติดมาด้วย ช่องว่างความรับผิดชอบของบทที่ 06 กลับมาอีกครั้งในรูปของการโทษที่กระจายไปแบบเดียวกับที่ความไว้วางใจกระจาย ความตึงเครียดระหว่าง stack กับ state ของ

> บทที่ 07 คมชัดขึ้นทันทีที่เงินเองไม่ต้องพึ่งรัฐอีกต่อไป ความแน่นอนในฐานะผลผลิตเชิงวิศวกรรมของบทที่ 12 กลับมาอีกครั้งในเกณฑ์หนึ่งในสามและเพดานที่พิสูจน์แล้วของความรู้ร่วม

Open questions · คำถามที่ยังเปิดอยู่

If perfect common knowledge is provably unreachable over any unreliable channel, is ordinary human coordination — running on language and gesture, channels no more reliable — merely tolerating the same gap with a longer patience for uncertainty? Does a system that formally guarantees agreement on facts quietly redefine "trust" down to "agreement," discarding what the word carried for Hobbes? And if no one — not even a quorum — is identifiable behind a permissionless ledger, whose sovereignty has actually been replaced?

> ถ้าความรู้ร่วมที่สมบูรณ์พิสูจน์ได้ว่าไปไม่ถึงผ่านช่องทางที่ไม่น่าเชื่อถือใดๆ เลย การประสานงานของมนุษย์ทั่วไป — ที่รันอยู่บนภาษาและท่าทาง ช่องทางที่ไม่ได้น่าเชื่อถือไปกว่ากันเลย — เป็นแค่การทนอยู่กับช่องว่างเดียวกันนั้นด้วยความอดทนต่อความไม่แน่นอนที่ยาวนานกว่าเท่านั้นหรือไม่ ระบบที่รับประกันข้อตกลงเรื่องข้อเท็จจริงอย่างเป็นรูปนัย กำลังนิยาม "trust" ใหม่อย่างเงียบๆ ให้เหลือแค่ "agreement" ทิ้งสิ่งที่คำคำนี้เคยแบกไว้สำหรับ Hobbes หรือไม่ และถ้าไม่มีใครเลย — แม้แต่ quorum เอง — ระบุตัวตนได้เบื้องหลังบัญชีแยกประเภทแบบไม่ต้องขออนุญาต (permissionless) แล้วอธิปไตยของใครกันแน่ที่ถูกแทนที่ไปจริงๆ

Go deeper · อ่านต่อ

— Thomas Hobbes, *Leviathan*, 1651.

> Thomas Hobbes, Leviathan, 1651.

— Leslie Lamport, Robert Shostak & Marshall Pease, "The Byzantine Generals Problem," *ACM Transactions on Programming Languages and Systems* 4, no. 3 (July 1982): 382–401; Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 31 October 2008 — bitcoin.org/bitcoin.pdf.

> Leslie Lamport, Robert Shostak & Marshall Pease, "The Byzantine Generals Problem," ACM Transactions on Programming Languages and Systems 4, no. 3 (July 1982): 382–401; Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 31 October 2008 — bitcoin.org/bitcoin.pdf.

— *Common Knowledge*, Stanford Encyclopedia of Philosophy.

> Common Knowledge, Stanford Encyclopedia of Philosophy.

CHAPTER 15

The Computational Universe

เอกภพเชิงคำนวณ

This is Part II's closing hearing: computation, having examined every other branch of philosophy, finally asks whether reality itself will take the stand — and here the two courts run out of ground to stand on separately.

นี่คือการไต่สวนปิดท้ายของภาค II: การคำนวณ หลังจากไต่สวนทุกสาขาอื่นของปรัชญามาแล้ว ในที่สุดก็ถามว่าความจริงเองจะขึ้นให้การหรือไม่ — และตรงนี้เองที่สองศาลหมดพื้นให้ยืนแยกจากกันอีกต่อไป

Is reality itself on the witness stand?

เอกภพเองอยู่ในที่นั่งพยานหรือไม่

On 2 June 2016, at Recode's Code Conference, Elon Musk told the audience there was "a one in billions chance we're in base reality," reasoning that video games had gone from Pong to photorealism in forty years, so a sufficiently advanced civilization would run simulations indistinguishable from the world we take for granted. The claim traces to Nick Bostrom's 2003 paper "Are You Living in a Computer Simulation?" — but Bostrom's actual argument is not that claim. It is a trilemma: at least one of the following is true — almost all civilizations go extinct before they can run high-fidelity ancestor simulations; those that survive are almost universally uninterested in running them; or we are almost certainly in one (Bostrom, *Philosophical Quarterly* 53, no. 211, 2003: 243–255). Bostrom assigns no precise probability to the third branch and warns explicitly against overconfidence in any of the three. A careful philosophical argument, built to be cross-examined by

philosophy, was cross-examined instead by a technology conference, and lost precision in the exchange. That collapse — trilemma into slogan — is itself worth entering as evidence before the argument proper begins.

> วันที่ 2 มิถุนายน 2016 ที่งาน Code Conference ของ Recode Elon Musk บอกผู้ฟังว่ามี "โอกาสหนึ่งในพันล้านที่เราอยู่ในความจริงฐาน" (a one in billions chance we're in base reality) โดยให้เหตุผลว่าวิดีโอเกมพัฒนาจาก Pong ไปถึงภาพสมจริงระดับภาพถ่ายได้ภายในสี่สิบปี ฉะนั้นอารยธรรมที่ก้าวหน้าพอย่อมรันการจำลองที่แยกไม่ออกจากโลกที่เราถือเป็นเรื่องปกติได้ ข้อกล่าวอ้างนี้สืบย้อนไปถึงเปเปอร์ปี 2003 ของ Nick Bostrom เรื่อง "Are You Living in a Computer Simulation?" — แต่ข้อโต้แย้งจริงของ Bostrom ไม่ใช่ข้อกล่าวอ้างนั้น มันคือ trilemma: อย่างน้อยหนึ่งในสามข้อต่อไปนี้เป็นจริง — อารยธรรมเกือบทั้งหมดสูญพันธุ์ก่อนจะรันการจำลองบรรพบุรุษที่ละเอียดสมจริงได้ หรือไม่ก็อารยธรรมที่รอดมาได้แทบทั้งหมดไม่สนใจจะรันมัน หรือไม่ก็เราแทบจะแน่นอนว่าอยู่ในการจำลองหนึ่งอยู่แล้ว (Bostrom, Philosophical Quarterly 53, no. 211, 2003: 243–255) Bostrom ไม่ได้ระบุความน่าจะเป็นที่แม่นยำให้กับข้อที่สาม และเตือนไว้อย่างชัดเจนไม่ให้มั่นใจเกินไปในข้อไหนทั้งสามข้อ ข้อโต้แย้งเชิงปรัชญาที่รอบคอบ ซึ่งสร้างมาให้ถูกซักถามค้านโดยปรัชญา กลับถูกซักถามค้านโดยงานประชุมด้านเทคโนโลยีแทน และสูญเสียความแม่นยำไปในการแลกเปลี่ยนนั้น การพังทลายนั้น — จาก trilemma กลายเป็นสโลแกน — คุ้มค่าที่จะนำเข้าเป็นหลักฐานเองก่อนที่ข้อโต้แย้งตัวจริงจะเริ่มต้น

Stated as Bostrom wrote it, the argument is a disjunction reached by elimination, hedged at every step: it does not say what we are, only that one of three things must be. Held to that standard, it is a piece of careful anthropic reasoning about future computing capacity — not a metaphysics, and not proof of anything about the world we are actually in.

> ตามที่ Bostrom เขียนไว้จริง ข้อโต้แย้งนี้คือ disjunction ที่ได้มาจากการตัดตัวเลือกออกทีละข้อ ระมัดระวังทุกขั้นตอน: มันไม่ได้บอกว่าเราเป็นอะไร บอกแค่ว่าหนึ่งในสามอย่างต้องเป็นจริง เมื่อยึดตามมาตรฐานนั้น มันคือชิ้นส่วนของการให้เหตุผลแบบ anthropic ที่รอบคอบเกี่ยวกับศักยภาพการคำนวณในอนาคต — ไม่ใช่อภิปรัชญา และไม่ใช่บทพิสูจน์อะไรเกี่ยวกับโลกที่เราอยู่จริงๆ

The proposal that physical law is itself computation runs from Konrad Zuse's *Rechnender Raum* (1969; translated *Calculating Space*, MIT, 1970), which modeled space as a cellular automaton computing its own next state, through Edward Fredkin's Finite Nature hypothesis — that space, time, and every physical quantity are ultimately discrete and finite — to Seth Lloyd's *Programming the Universe* (2006), which treats every particle collision as a bit-flip, and to the Wolfram Physics Project, launched 14 April 2020, which models space as an evolving hypergraph from which physical law emerges. The skepticism deserves equal floor, not a footnote. Sabine Hossenfelder has argued that no one has shown how a discrete algorithm reproduces the continuous symmetries of general relativity, and that searches for a preferred discretization scale have found nothing (Hossenfelder, "The Simulation Hypothesis Is Pseudoscience," *Backreaction*, February 2021). Scott Aaronson makes the sharper point: the hypothesis's problem is not that it has been shown false, but that essentially no observation could count against it — a defect prior to any question of truth ("Because you asked: the Simulation Hypothesis has not been falsified; remains unfalsifiable," *Shtetl-Optimized*, 3 October 2017).

> ข้อเสนอที่ว่ากฎฟิสิกส์เองก็คือการคำนวณ ไล่มาตั้งแต่ Rechnender Raum ของ Konrad Zuse (1969; แปลเป็น Calculating Space, MIT, 1970) ซึ่งจำลองอวกาศเป็นเซลลูลาร์ออโตมาตา (cellular automaton) ที่คำนวณสถานะถัดไปของตัวเอง ผ่านสมมติฐาน Finite Nature ของ Edward Fredkin — ที่ว่าอวกาศ เวลา และปริมาณทางฟิสิกส์ทุกอย่างท้ายที่สุดแล้วไม่ต่อเนื่องและมีจำกัด — ไปจนถึง Programming the Universe (2006) ของ Seth Lloyd ที่มองการชนกันของอนุภาคทุกครั้งเป็นการพลิกบิตหนึ่งบิต และไปจนถึง Wolfram Physics Project ที่เปิดตัวเมื่อวันที่ 14 เมษายน 2020 ซึ่งจำลองอวกาศเป็นไฮเปอร์กราฟที่วิวัฒน์ไปเรื่อยๆ อันเป็นที่มาของกฎฟิสิกส์ ความสงสัยควรได้พื้นที่เท่ากัน ไม่ใช่แค่เชิงอรรถ Sabine Hossenfelder โต้แย้งว่าไม่มีใครแสดงให้เห็นได้เลยว่าอัลกอริทึมไม่ต่อเนื่องจะจำลองสมมาตรต่อเนื่องของทฤษฎีสัมพัทธภาพทั่วไปได้อย่างไร และการค้นหาสเกลการแยกไม่ต่อเนื่องที่เหมาะสมก็ไม่พบอะไรเลย (Hossenfelder, "The Simulation Hypothesis Is Pseudoscience," Backreaction, February 2021) Scott Aaronson ชี้ประเด็นที่คมกว่านั้น: ปัญหาของสมมติฐานนี้ไม่ใช่ว่ามันถูกแสดงว่าเป็นเท็จ แต่

> แทบไม่มีการสังเกตใดเลยที่จะนับเป็นหลักฐานค้านมันได้ — ข้อบกพร่องที่เกิดก่อนคำถามเรื่องความจริงใดๆ จะเริ่มด้วยซ้ำ ("Because you asked: the Simulation Hypothesis has not been falsified; remains unfalsifiable," Shtetl-Optimized, 3 October 2017)

John Conway's Game of Life, publicized by Martin Gardner in *Scientific American* in October 1970, runs on three rules and produces gliders, oscillators, and — decades later — a complete computer built from its own pieces. Rule 110, one of 256 possible elementary one-dimensional automata, was conjectured universal by Stephen Wolfram in 1985 and proved so by Matthew Cook, who worked out the proof while bound by a nondisclosure agreement with Wolfram Research that delayed its publication roughly a decade ("Universality in Elementary Cellular Automata," *Complex Systems* 15, no. 1, 2004: 1–40). Both cases make the same point without metaphor: full computational power, and the unpredictability that comes with it, can hide inside a rule simple enough to state in one sentence.

> Game of Life ของ John Conway ที่ Martin Gardner เผยแพร่ใน Scientific American เมื่อเดือนตุลาคม 1970 รันด้วยกฎแค่สามข้อ และผลิต glider, oscillator และ — หลายสิบปีต่อมา — คอมพิวเตอร์สมบูรณ์แบบที่สร้างขึ้นจากชิ้นส่วนของมันเอง Rule 110 หนึ่งใน 256 elementary one-dimensional automata ที่เป็นไปได้ ถูก Stephen Wolfram คาดการณ์ไว้ในปี 1985 ว่าเป็น universal และถูกพิสูจน์จริงโดย Matthew Cook ผู้คิดค้นบทพิสูจน์นี้ขึ้นระหว่างที่ถูกผูกมัดด้วยสัญญาไม่เปิดเผยข้อมูลกับ Wolfram Research ซึ่งทำให้การตีพิมพ์ล่าช้าไปราวหนึ่งทศวรรษ ("Universality in Elementary Cellular Automata," Complex Systems 15, no. 1, 2004: 1–40) ทั้งสองกรณีชี้ประเด็นเดียวกันโดยไม่ต้องใช้อุปมาเลย: พลังการคำนวณเต็มรูปแบบ พร้อมกับความคาดเดาไม่ได้ที่ติดมาด้วย ซ่อนอยู่ในกฎที่ง่ายพอจะพูดจบในประโยคเดียวได้

Every "random" number a computer produces is the deterministic output of an algorithm seeded by some prior state; it looks unpredictable only because the algorithm is opaque to whoever is asking. This restages compatibilism's oldest move as engineering. Hobbes and Hume, the classical

compatibilists, held that freedom never meant *uncaused* — it meant acting from one's own will, unconstrained by external force — so a fully determined pseudo-random generator and a fully determined human can both exhibit exactly the unpredictability-to-an-observer that ordinary judgments of freedom actually track. If physical law itself turns out to be the deterministic output of a cellular automaton, that does not resolve the question; it removes the last place to hide from having to answer it.

> ตัวเลข "สุ่ม" ทุกตัวที่คอมพิวเตอร์ผลิตออกมา คือผลลัพธ์ที่ดีเทอร์มินิสติกของอัลกอริทึมที่ตั้งค่าเริ่มต้นจากสถานะก่อนหน้าบางอย่าง มันดูคาดเดาไม่ได้ก็เพราะอัลกอริทึมนั้นทึบสำหรับใครก็ตามที่กำลังถาม เรื่องนี้เล่นท่าเก่าแก่ที่สุดของประนีประนอมนิยม (compatibilism) ขึ้นมาใหม่ในรูปวิศวกรรม Hobbes และ Hume สอง compatibilist คลาสสิก ยืนยันว่าเสรีภาพไม่เคยแปลว่า ไม่มีเหตุ — มันแปลว่าการกระทำจากเจตจำนงของตัวเอง โดยไม่ถูกบังคับจากแรงภายนอก — ฉะนั้นทั้งตัวสุ่มเทียมที่ถูกกำหนดไว้เต็มรูปแบบ และมนุษย์ที่ถูกกำหนดไว้เต็มรูปแบบต่างก็แสดงความคาดเดาไม่ได้-ต่อผู้สังเกตแบบเดียวกันเป๊ะๆ กับที่การตัดสินเรื่องเสรีภาพตามปกติใช้วัดอยู่จริง ถ้ากฎฟิสิกส์เองปรากฏว่าเป็นผลลัพธ์ดีเทอร์มินิสติกของ cellular automaton ตัวหนึ่ง นั่นก็ไม่ได้แก้คำถามนี้เลย มันแค่ตัดที่หลบซ่อนสุดท้ายออกไป จากการต้องตอบคำถามนั้นเท่านั้น

Every principle above runs the same shape, and the shape is the turn this book keeps making: a computational claim meets a question philosophy has held open for centuries, and neither side gets the last word. Bostrom's trilemma stays open; Hossenfelder's objection stays unanswered; compatibilism stays contested. That is not this method failing to reach a verdict. It is what an honest cross-examination looks like when the witness is reality itself, and the trial is still open when the file closes.

> ทุกหลักการข้างต้นวิ่งตามรูปแบบเดียวกัน และรูปแบบนั้นคือลีลาที่หนังสือเล่มนี้ทำซ้ำอยู่เสมอ: ข้อกล่าวอ้างเชิงคำนวณมาเจอกับคำถามที่ปรัชญาเปิดค้างไว้มานานหลายศตวรรษ และไม่มีฝ่ายไหนได้คำพูดสุดท้าย trilemma ของ Bostrom ยังเปิดอยู่

> ข้อคัดค้านของ Hossenfelder ยังไม่มีคำตอบ compatibilism ยังถูกโต้แย้งอยู่ นั่นไม่ใช่วิธีการนี้ล้มเหลวที่จะไปถึงคำตัดสิน แต่มันคือหน้าตาของการซักถามค้านที่ซื่อตรงเมื่อพยานคือความจริงเองล้วนๆ และการพิจารณาคดียังเปิดอยู่ตอนที่แฟ้มถูกปิดลง

The people, briefly. Bostrom, a Swedish-born philosopher at Oxford, wrote the 2003 paper after first circulating it in 2001. Zuse, better known for building one of the first working programmable computers in the 1930s and 1940s, turned to digital physics afterward. Fredkin worked the idea for decades until his death in 2023. Lloyd is a mechanical-engineering professor at MIT and an architect of quantum computing. Wolfram, having built Mathematica, launched his physics project as an open, crowd-participation effort in 2020. Hossenfelder is a German theoretical physicist known for public skepticism toward untestable claims dressed as physics. Conway devised Life in 1970; Cook proved Rule 110 universal and then spent years fighting to publish it.

> **ผู้คน โดยย่อ** Bostrom นักปรัชญาเชื้อสายสวีเดนที่ Oxford เขียนเปเปอร์ปี 2003 หลังจากเผยแพร่ฉบับร่างครั้งแรกในปี 2001 Zuse ซึ่งเป็นที่รู้จักมากกว่าจากการสร้างคอมพิวเตอร์ที่ตั้งโปรแกรมได้เครื่องแรกๆ ที่ใช้งานได้จริงในทศวรรษ 1930 และ 1940 หันมาสนใจฟิสิกส์เชิงดิจิทัล (digital physics) ในภายหลัง Fredkin ทำงานกับไอเดียนี้มาหลายทศวรรษจนกระทั่งเสียชีวิตในปี 2023 Lloyd เป็นศาสตราจารย์ด้านวิศวกรรมเครื่องกลที่ MIT และเป็นหนึ่งในสถาปนิกของ quantum computing Wolfram หลังจากสร้าง Mathematica แล้ว เปิดตัวโครงการฟิสิกส์ของเขาเป็นความพยายามแบบเปิดให้คนทั่วไปร่วมมือในปี 2020 Hossenfelder เป็นนักฟิสิกส์ทฤษฎีชาวเยอรมันที่เป็นที่รู้จักจากความสงสัยต่อสาธารณะต่อข้อกล่าวอ้างที่พิสูจน์ไม่ได้แต่แต่งตัวเป็นฟิสิกส์ Conway คิดค้น Life ขึ้นในปี 1970 Cook พิสูจน์ว่า Rule 110 เป็น universal แล้วใช้เวลาหลายปีต่อสู้เพื่อตีพิมพ์มัน

Connections · เชื่อมโยง

Chapter 02's question — are objects in a simulated world real — is this chapter's necessary predecessor. Chapter 09's Church–Turing thesis sets the limit on what "computing the universe" could even mean. Chapter 11's

audit of Shannon's bracket, and Wheeler's it-from-bit, is where the speculation in this chapter meets its own foundations. Chapter 10's question of what is reachable, not merely true, asks whether any simulator could compute reality at the resolution reality displays.

> คำถามของบทที่ 02 — วัตถุในโลกจำลองเป็นของจริงหรือไม่ — คือสิ่งที่บทนี้ต้องอาศัยเป็นพื้นมาก่อน ข้อตั้ง Church–Turing ของบทที่ 09 กำหนดขอบเขตว่า "การคำนวณเอกภพ" จะแปลว่าอะไรได้บ้างตั้งแต่ต้น การตรวจสอบวงเล็บของ Shannon ในบทที่ 11 และ it from bit (สสารจากบิต) ของ Wheeler คือจุดที่การคาดเดาในบทนี้มาเจอกับรากฐานของตัวมันเอง คำถามของบทที่ 10 เรื่องสิ่งที่ไปถึงได้ ไม่ใช่แค่สิ่งที่เป็นจริง ถามว่าตัวจำลองใดๆ จะคำนวณความจริงได้ที่ความละเอียดเท่าที่ความจริงแสดงออกมาได้หรือไม่

Open questions · คำถามที่ยังเปิดอยู่

If no experiment could ever distinguish a simulated universe from a "base" one, has the argument stopped being the kind of claim philosophy or physics can adjudicate at all? Does proving Rule 110 universal make Wolfram's hypergraph more likely to be physics, or only more likely to be capable of it, pending some independent reason to believe the update rule? And is compatibilism actually answering the free-will question, or quietly redefining "free" until the harder version of the question can no longer be asked?

> ถ้าไม่มีการทดลองใดเลยที่จะแยกเอกภพจำลองออกจากเอกภพ "ฐาน" ได้ ข้อโต้แย้งนี้หยุดเป็นข้อกล่าวอ้างชนิดที่ปรัชญาหรือฟิสิกส์ตัดสินได้เลยหรือไม่ การพิสูจน์ว่า Rule 110 เป็น universal ทำให้ไฮเปอร์กราฟของ Wolfram มีโอกาสเป็นฟิสิกส์มากขึ้นจริงหรือ หรือแค่มีโอกาสมีศักยภาพเป็นฟิสิกส์ได้มากขึ้นเท่านั้น รอเหตุผลอิสระบางอย่างที่จะเชื่อกฎการอัปเดตนั้น และ compatibilism กำลังตอบคำถามเรื่องเจตจำนงเสรี (free will) จริงๆ หรือแค่นิยาม "free" ใหม่อย่างเงียบๆ จนคำถามเวอร์ชันที่ยากกว่านั้นถูกถามไม่ได้อีกต่อไป

Go deeper · อ่านต่อ

— Seth Lloyd, *Programming the Universe: A Quantum Computer Scientist Takes On the Cosmos*, Knopf, 2006.

> Seth Lloyd, Programming the Universe: A Quantum Computer Scientist Takes On the Cosmos, Knopf, 2006.

— Nick Bostrom, "Are You Living in a Computer Simulation?", *Philosophical Quarterly* 53, no. 211 (2003): 243–255 — simulation-argument.com/simulation.pdf; Matthew Cook, "Universality in Elementary Cellular Automata," *Complex Systems* 15, no. 1 (2004): 1–40.

> Nick Bostrom, "Are You Living in a Computer Simulation?", Philosophical Quarterly 53, no. 211 (2003): 243–255 — simulation-argument.com/simulation.pdf; Matthew Cook, "Universality in Elementary Cellular Automata," Complex Systems 15, no. 1 (2004): 1–40.

— *Computer Simulations in Science*, Stanford Encyclopedia of Philosophy.

> Computer Simulations in Science, Stanford Encyclopedia of Philosophy.

CLOSING

Closing

ปิดเล่ม

This is the courtroom after both benches have been used — no verdict read out, only the record handed across and the door left unlocked.

นี่คือห้องพิจารณาคดีหลังม้านั่งคู่ความทั้งสองฝั่งถูกใช้งานแล้ว — ไม่มีการอ่านคำตัดสิน มีแต่บันทึกคดีที่ถูกส่งต่อ และประตูที่ถูกปล่อยไว้ไม่ล็อก

What was the trial for?

การพิจารณาคดีนี้มีไว้เพื่ออะไร

Count the collisions this book has entered into evidence. Church and Turing settled the *Entscheidungsproblem* six weeks apart in 1936 without either having seen the other's proof, by methods that had nothing in common until Turing wrote the appendix showing they agreed (ch.09). Stephen Cook located the boundary of feasible computation in 1971; Leonid Levin, cut off from Western journals, found the identical theorem two years later (ch.10). Haskell Curry, William Howard, and N. G. de Bruijn each arrived at the correspondence between proofs and programs from a different direction, none of them aware of the others (ch.12). George Boole rebuilt Leibniz's calculus of reasoning without knowing Leibniz had been there, and only learned of the anticipation after his book was printed (ch.01). Four independent rediscoveries of the same seam is not four coincidences. It is what a single vein looks like when people keep striking it from opposite sides of a wall.

> นับจำนวนการชนกันโดยบังเอิญที่หนังสือเล่มนี้นำเข้าเป็นหลักฐาน Church และ Turing แก้ Entscheidungsproblem ห่างกันแค่หกสัปดาห์ในปี 1936 โดยไม่มีใครเห็นบทพิสูจน์ของอีกฝ่ายเลย ด้วยวิธีที่ไม่มีอะไรร่วมกันเลย จนกระทั่ง Turing เขียนภาคผนวกแสดงให้เห็นว่าทั้งสองวิธีให้ผลตรงกัน (บทที่ 09) Stephen Cook หา

ขอบเขตของการคำนวณที่เป็นไปได้จริงเจอในปี 1971 Leonid Levin ซึ่งถูกตัดขาดจากวารสารตะวันตก พบทฤษฎีบทเดียวกันเป๊ะๆ ในอีกสองปีถัดมา (บทที่ 10) Haskell Curry, William Howard และ N. G. de Bruijn ต่างมาถึงความสอดคล้องระหว่างบทพิสูจน์กับโปรแกรมจากคนละทิศทาง โดยไม่มีใครรู้ว่าอีกสองคนกำลังทำอะไรอยู่ (บทที่ 12) George Boole สร้างแคลคูลัสแห่งเหตุผลของ Leibniz ขึ้นมาใหม่โดยไม่รู้ว่า Leibniz เคยไปถึงจุดนั้นมาก่อนแล้ว และรู้ว่ามีคนคาดการณ์ไว้ก่อนก็หลังจากหนังสือของเขาถูกพิมพ์ออกไปแล้ว (บทที่ 01) การค้นพบซ้ำโดยอิสระต่อกันสี่ครั้งของรอยเดียวกันไม่ใช่เรื่องบังเอิญสี่ครั้ง มันคือหน้าตาของแร่สายเดียวที่คนเจาะเข้าไปหาจากคนละฝั่งของกำแพงอยู่เรื่อยๆ

That is the whole claim, and it is why this book was built as a cross-examination rather than a survey. Philosophy and computing are one project — *what can be known, said, and done by rule* — separated by an accident of institutional history rather than by a difference in subject. Part I put the machine on the stand and let philosophy ask what a program is, whether a machine's answer counts as knowledge, how code means, whether a delegated decision keeps its morality, and who governs when architecture outruns law. Part II reversed the benches, and computation's questions were the more embarrassing ones: your ideal knower is computationally impossible (ch.10); your certainty is an engineering product with an error budget (ch.12); your account of trust is a threshold with a number on it (ch.14).

นั่นคือข้อกล่าวอ้างทั้งหมดของหนังสือเล่มนี้ และเป็นเหตุผลว่าทำไมมันถึงถูกสร้างขึ้นเป็นการซักถามค้าน ไม่ใช่การสำรวจ ปรัชญากับการคำนวณคือโครงการเดียวกัน — สิ่งที่รู้ได้ พูดได้ และทำได้ตามกฎ — ที่แยกจากกันด้วยอุบัติเหตุของประวัติศาสตร์เชิงสถาบัน ไม่ใช่ด้วยความต่างของหัวข้อ ภาค I เอาเครื่องจักรขึ้นให้การ และให้ปรัชญาถามว่าโปรแกรมคืออะไร คำตอบของเครื่องนับเป็นความรู้หรือไม่ โค้ดมีความหมายได้อย่างไร การตัดสินใจที่ถูกมอบหมายให้ทำแทนยังรักษาศีลธรรมของมันไว้ได้หรือไม่ และใครปกครองเมื่อสถาปัตยกรรมวิ่งแซงหน้ากฎหมาย ภาค II สลับม้านั่งคู่ความ และคำถามของฝั่งการคำนวณกลับเป็นคำถามที่น่าอายกว่า: ผู้รู้ใน

> อุดมคติของปรัชญาเป็นไปไม่ได้เชิงการคำนวณ (บทที่ 10) ความแน่นอนของปรัชญาคือผลผลิตเชิงวิศวกรรมที่มีงบประมาณความผิดพลาด (บทที่ 12) คำอธิบายเรื่องความไว้วางใจของปรัชญาคือเกณฑ์ที่มีตัวเลขกำกับอยู่ (บทที่ 14)

Neither court won, and that is not a failure of the method. Every hearing here ends the way an honest examination ends: with the claim narrower than it was, and the ground under it visible. Chapter 15 said why the trial cannot close. This chapter says who runs it next.

> ไม่มีศาลไหนชนะ และนั่นไม่ใช่ความล้มเหลวของวิธีการ ทุกการไต่สวนในเล่มนี้จบลงแบบเดียวกับที่การซักถามที่ซื่อตรงจบลงเสมอ: ด้วยข้อกล่าวอ้างที่แคบลงกว่าเดิมและพื้นดินใต้มันที่มองเห็นได้ บทที่ 15 บอกแล้วว่าทำไมการพิจารณาคดีถึงปิดไม่ได้ บทนี้บอกว่าใครเป็นผู้ดำเนินการมันต่อไป

The route · เส้นทาง

Thirty days, one hour a day. Read the book's chapter first, then the source; the encyclopedia entries are load-bearing, not decoration, and every link below was checked before this file shipped. Where a chapter gets two days, spend the second one arguing with the first.

> สามสิบวัน วันละหนึ่งชั่วโมง อ่านบทของหนังสือก่อน แล้วค่อยอ่านแหล่งที่มา รายการสารานุกรมเป็นส่วนที่แบกน้ำหนักจริง ไม่ใช่ของประดับ และลิงก์ทุกอันด้านล่างถูกตรวจสอบแล้วก่อนไฟล์นี้จะถูกส่งออกไป บทไหนที่ได้สองวัน ให้ใช้วันที่สองถกเถียงกับวันแรก

Days	Chapter	Start with	Then
1–2	00 The Map	SEP, Philosophy of Computer Science	SEP, Hilbert's Program
3–4	01 Logic	SEP, Gödel's Incompleteness Theorems	SEP, Leibniz's Influence on 19th Century Logic
5–6	02 Metaphysics	SEP, Types and Tokens	re-read ch.02's wall argument against ch.04

Days	Chapter	Start with	Then
7–8	03 Epistemology	SEP, Computer Simulations in Science	Robertson, Sanders, Seymour & Thomas, The Four Color Theorem
9–10	04 Mind	SEP, The Computational Theory of Mind · The Chinese Room Argument	Turing, Computing Machinery and Intelligence (1950)
11–12	05 Language	SEP, Speech Acts	SEP, Montague Semantics
13–14	06 Ethics	SEP, Ethics of AI and Robotics	Kleinberg, Mullainathan & Raghavan, Inherent Trade-Offs
15–16	07 Politics	SEP, Political Legitimacy	SEP, Privacy and Information Technology
17	08 Aesthetics	SEP, Martin Heidegger	Dreyfus, Alchemy and Artificial Intelligence (1965)
18	*the benches reverse*	re-read ch.00's three rules of examination	write down which side you have been quietly siding with
19–20	09 Computability	SEP, The Church-Turing Thesis	ch.09's open questions, answered in your own words
21–22	10 Complexity	SEP, Computational Complexity Theory	Aaronson, Why Philosophers Should Care (2011)
23–24	11 Information	SEP, Information	ch.11's audit, re-run against ch.00's Shannon bracket
25–26	12 Types & Proofs	SEP, Intuitionism · Intuitionistic Type Theory	Girard, Lafont & Taylor, *Proofs and Types*
27–28	13 Learning Machines	SEP, Rationalism vs. Empiricism · The Problem of Induction	Sutton, The Bitter Lesson (2019)
29	14 Distributed Trust	SEP, Common Knowledge	Nakamoto, Bitcoin (2008)

Days	Chapter	Start with	Then
30	15 The Computational Universe	Bostrom, Are You Living in a Computer Simulation? (2003)	pick the branch of the trilemma you reject, and say why

วัน	บท	เริ่มจาก	ต่อด้วย
1–2	00 แผนที่	SEP, Philosophy of Computer Science	SEP, Hilbert's Program
3–4	01 ตรรกศาสตร์	SEP, Gödel's Incompleteness Theorems	SEP, Leibniz's Influence on 19th Century Logic
5–6	02 อภิปรัชญา	SEP, Types and Tokens	อ่านข้อโต้แย้งเรื่องกำแพงของบทที่ 02 ซ้ำ เทียบกับบทที่ 04
7–8	03 ญาณวิทยา	SEP, Computer Simulations in Science	Robertson, Sanders, Seymour & Thomas, The Four Color Theorem
9–10	04 ปรัชญาแห่งจิต	SEP, The Computational Theory of Mind · The Chinese Room Argument	Turing, Computing Machinery and Intelligence (1950)
11–12	05 ปรัชญาภาษา	SEP, Speech Acts	SEP, Montague Semantics
13–14	06 จริยศาสตร์	SEP, Ethics of AI and Robotics	Kleinberg, Mullainathan & Raghavan, Inherent Trade-Offs
15–16	07 ปรัชญาการเมือง	SEP, Political Legitimacy	SEP, Privacy and Information Technology
17	08 สุนทรียศาสตร์และปรากฏการณ์วิทยา	SEP, Martin Heidegger	Dreyfus, Alchemy and Artificial Intelligence (1965)
18	ม้านั่งสลับฝั่ง	อ่านกฎสามข้อของการซักถามจากบทที่ 00 ซ้ำ	เขียนบันทึกว่าที่ผ่านมาเอนเข้าข้างฝ่ายไหนอยู่อย่างเงียบๆ
19–20	09 การคำนวณได้	SEP, The Church-Turing Thesis	คำถามที่ยังเปิดอยู่ของบทที่ 09 ตอบด้วยคำพูดของตัวเอง

วัน	บท	เริ่มจาก	ต่อด้วย
21–22	10 ความซับซ้อน	SEP, Computational Complexity Theory	Aaronson, Why Philosophers Should Care (2011)
23–24	11 สารสนเทศ	SEP, Information	การตรวจสอบของบทที่ 11 รันซ้ำเทียบกับวงเล็บของ Shannon จากบทที่ 00
25–26	12 ชนิดกับบทพิสูจน์	SEP, Intuitionism · Intuitionistic Type Theory	Girard, Lafont & Taylor, Proofs and Types
27–28	13 เครื่องเรียนรู้	SEP, Rationalism vs. Empiricism · The Problem of Induction	Sutton, The Bitter Lesson (2019)
29	14 ความไว้วางใจแบบกระจาย	SEP, Common Knowledge	Nakamoto, Bitcoin (2008)
30	15 เอกภพเชิงคำนวณ	Bostrom, Are You Living in a Computer Simulation? (2003)	เลือกแขนงหนึ่งของ trilemma ที่ปฏิเสธ แล้วบอกว่าทำไม

Two habits matter more than the schedule. Read one primary source per chapter rather than three summaries of it — the summaries agree with each other and the primary does not, and the disagreement is where the thinking is. And keep the last column: the exercises are the examination, and a route with no cross-examination in it is a reading list, not a training.

> มีนิสัยสองอย่างที่สำคัญกว่าตารางเวลานี้เสียอีก อ่านแหล่งปฐมภูมิ (primary source) หนึ่งชิ้นต่อบท แทนที่จะอ่านบทสรุปสามชิ้นของมัน — บทสรุปเห็นตรงกันหมด แต่แหล่งปฐมภูมิไม่เห็นด้วย และความไม่ลงรอยกันนั้นแหละคือจุดที่การคิดเกิดขึ้นจริง และรักษาคอลัมน์สุดท้ายไว้ให้ดี: แบบฝึกหัดคือการซักถาม และเส้นทางที่ไม่มีการซักถามค้านอยู่ในตัวมันเลย ก็เป็นแค่รายการหนังสืออ่าน ไม่ใช่การฝึกฝน

The license · ใบอนุญาต

Nothing in this book required a degree in either discipline to produce, and the record says so. Boole was a schoolmaster in Lincoln when he algebraized logic. Shannon was a twenty-one-year-old master's student when he showed that Boole's algebra was also a wiring diagram. Turing wrote the founding paper of computing for a mathematics journal and the founding question of machine minds for a philosophy journal, fourteen years apart, without changing fields — because there was no field to change. The department that separated them was founded in 1962, twenty-six years after the paper that made it possible. An institutional boundary that young has no authority over a reader.

> ไม่มีส่วนไหนในหนังสือเล่มนี้ที่ต้องใช้ปริญญาในสาขาใดสาขาหนึ่งเพื่อผลิตมันขึ้นมา และบันทึกคดีก็ยืนยันแบบนั้น Boole เป็นครูใหญ่โรงเรียนที่ Lincoln ตอนที่เขาทำให้ตรรกะกลายเป็นพีชคณิต Shannon เป็นนักศึกษาปริญญาโทอายุยี่สิบเอ็ดปีตอนที่เขาแสดงให้เห็นว่าพีชคณิตของ Boole ก็เป็นแผนภาพวงจรไฟฟ้าได้เหมือนกัน Turing เขียนเปเปอร์ก่อตั้งวงการการคำนวณให้วารสารคณิตศาสตร์ และคำถามก่อตั้งเรื่องจิตของเครื่องจักรให้วารสารปรัชญา ห่างกันสิบสี่ปี โดยไม่ได้เปลี่ยนสาขาเลยสักครั้ง — เพราะไม่มีสาขาให้เปลี่ยน ภาควิชาที่แยกทั้งสองออกจากกันก่อตั้งขึ้นในปี 1962 ยี่สิบหกปีหลังเปเปอร์ที่ทำให้มันเป็นไปได้ พรมแดนเชิงสถาบันที่อายุน้อยขนาดนั้นไม่มีอำนาจเหนือผู้อ่านคนไหนเลย

That is the license, and it is narrower and better than permission to have opinions. It is permission to ask a question in the other discipline's vocabulary and to insist on an answer there: to ask an engineer what their system means and not accept a benchmark score, to ask a philosopher what their ideal agent would cost to run and not accept that the question is vulgar. Each field's habits protect it from exactly one kind of question, and it is always the other field's question.

> นั่นคือใบอนุญาต และมันแคบกว่าและดีกว่าการอนุญาตให้มีความเห็นเสียอีก มันคือการอนุญาตให้ถามคำถามด้วยคำศัพท์ของอีกสาขาหนึ่ง และยืนกรานเอาคำตอบจากตรงนั้นให้ได้: ถามวิศวกรว่าระบบของพวกเขาหมายความว่าอะไร แล้วไม่ยอมรับแค่

> คะแนน benchmark ถามนักปรัชญาว่าตัวแทนในอุดมคติของพวกเขาจะมีต้นทุนรันเท่าไร แล้วไม่ยอมรับว่าคำถามนั้นหยาบคาย นิสัยของแต่ละสาขาปกป้องตัวเองจากคำถามชนิดเดียวเป๊ะๆ และมันก็เป็นคำถามของอีกสาขาหนึ่งเสมอ

The rules were set out in chapter 00 and they are the only equipment needed. The examiner sets the terms. Both voices appear, or it is a lecture. Every claim carries its evidence — a date, a work, a source that can be opened — because the fastest way to lose a cross-examination is to assert what the record will not support.

> กฎเหล่านี้ถูกวางไว้ในบทที่ 00 แล้ว และมันคืออุปกรณ์เดียวที่จำเป็นต้องใช้ ผู้ซักถามเป็นผู้กำหนดเงื่อนไข ทั้งสองเสียงต้องปรากฏ ไม่อย่างนั้นมันก็แค่การบรรยาย ทุกข้อกล่าวอ้างต้องแบกหลักฐานของมันไว้ — วันที่ ผลงานชิ้นหนึ่ง แหล่งที่มาที่เปิดดูได้จริง — เพราะทางที่เร็วที่สุดที่จะแพ้การซักถามค้านคือการยืนยันสิ่งที่บันทึกคดีไม่รองรับ

The record is now yours. The witnesses are still under oath, the questions in every chapter's *Open questions* were left open on purpose, and the chair at the front of the room has been empty since chapter 15. Sit down and begin.

> บันทึกคดีเป็นของคุณแล้ว พยานทุกคนยังอยู่ภายใต้คำสาบานอยู่ คำถามใน คำถามที่ยังเปิดอยู่ ของทุกบทถูกปล่อยให้เปิดค้างไว้โดยตั้งใจ และเก้าอี้ที่หัวห้องว่างเปล่ามาตั้งแต่บทที่ 15 นั่งลง แล้วเริ่ม

The Examined Machine · the AICE Library, 2026.

ไม่มีคำตัดสินในเล่มนี้ มีแต่บันทึกคดีที่ส่งต่อมาถึงมือคุณแล้ว