HISTORY OF AI

The Birth & Growth of Real AI

Turing & the Imitation Game · The Dartmouth Workshop

Symbolic AI & GOFAI · The First AI Winter

Expert Systems · The Second AI Winter

Philip Osztromok

Generated with Claude

Table of Contents

History of AI — The Birth & Growth of Real AI — 8 Chapters

🧮 Turing & the Imitation Game

Course 1 covered nothing but stories, hoaxes, and fiction — imagined machines with no engineering behind them at all. This chapter is the actual pivot point of this entire "History of AI" series: a real mathematician, taking the question "can machines think?" seriously enough to turn it into something testable, rather than something to argue about in the abstract forever.

Who Was Asking

Alan Turing was already, by 1950, one of the most consequential figures in the early history of computing — his 1936 concept of the "Turing machine" underpins the theory of computation itself, and his codebreaking work at Bletchley Park during the Second World War had already demonstrated, in practice, what a machine built to process information systematically could achieve under real pressure. He wasn't a science-fiction writer speculating about the future; he was one of the people actually building the mathematical foundations the future of computing would run on.

📄 "Computing Machinery and Intelligence" (1950)

Turing's paper, published in the philosophy journal *Mind*, opens with a now-famous line, and an immediate pivot:

> **The Opening Move**
>
> > "I propose to consider the question, 'Can machines think?'"
>
> Turing then argues the question itself is too poorly defined to be useful — "thinking" and "machine" are both words loaded with assumptions nobody agrees on. Rather than trying to settle an unanswerable definitional argument, he proposes replacing it entirely with a different, deliberately more concrete question, framed as a game.

🎭 The Imitation Game

Turing's test began as a variation on an existing parlor game, in which a hidden man and woman try to convince an interrogator (communicating only via written notes) of their gender, with the woman trying to help the interrogator guess correctly and the man trying to deceive them. Turing's adaptation replaces one of the participants with a machine:

> The Setup
>
> A human interrogator communicates via text (avoiding any voice or visual cues) with two hidden participants — one human, one machine — without knowing which is which. Both try to convince the interrogator they're the human. If a machine can regularly succeed at this — fooling the interrogator no more rarely than an actual human competitor would — Turing proposes we should be willing to say it "thinks," at least for practical purposes.

This is worth being precise about, since it's frequently oversimplified in popular retellings: Turing isn't claiming a machine that passes this test definitely possesses genuine inner consciousness or understanding — he's proposing that the question "can it think?" be *replaced* by the more answerable question "can it convincingly imitate a thinking being?", on the grounds that the original question may be unanswerable in principle, while this one, at least, can actually be tested.

🛡️ Turing Pre-Empted His Critics

A substantial portion of the paper is Turing anticipating and directly rebutting objections he expected — a genuinely unusual and rigorous move for a 1950 paper on a topic this speculative. Several are still cited by name today:

Objection	Turing's Rebuttal (in brief)
Theological Objection — thinking requires a soul, and only humans/animals have one	Treats this as a matter of faith, not evidence, and questions why a soul couldn't be granted to a machine if one exists at all
"Heads in the Sand" Objection — the consequences would be too dreadful, so it must be false	Notes this is a wish, not an argument
Mathematical Objection — Gödel's incompleteness theorems show formal systems have inherent limits	Argues human minds may have exactly the same limits, so this doesn't prove machines are uniquely constrained
Lady Lovelace's Objection — a machine can only do what it's explicitly programmed to do, never anything original	Argues a sufficiently complex machine could produce genuinely surprising behavior its programmers didn't anticipate

⚠ Lady Lovelace's Objection Is Still Argued About Today

The objection is named for Ada Lovelace, who wrote in 1843 (regarding Charles Babbage's proposed Analytical Engine) that a machine "has no pretensions to originate anything" and can only do whatever it is explicitly ordered to perform. Turing's rebuttal — that a complex enough system could surprise even its own creators — is a genuinely live argument in modern discussions of machine learning systems, which frequently do produce outputs and behaviors their own designers didn't specifically anticipate or intend, for better and for worse.

♟️ A Direct Answer to Course 1's Central Question

🔮 This Is the Mechanical Turk Question, Taken Seriously

Course 1 spent an entire chapter on the Mechanical Turk fooling audiences for 80 years through a convincing performance, with no genuine mechanism behind it at all. Turing's test takes that exact tension — a convincing performance vs. genuine underlying capability — and instead of treating the gap as an embarrassing flaw to expose, proposes taking the performance itself seriously as the actual definition worth testing. It's a genuinely bold philosophical move: rather than asking "is this real, or is this a trick?", Turing asks "if we genuinely can't tell the difference, does the distinction still matter?"

📜 Legacy and Later Criticism

The Turing Test became — and remains — one of the most cited ideas in the entire history of AI, frequently invoked (correctly or not) as a benchmark for machine intelligence. It has also drawn serious philosophical criticism over the decades, most famously philosopher John Searle's 1980 "Chinese Room" thought experiment, which argues that convincingly manipulating symbols according to rules (exactly what Turing's test measures) doesn't necessarily mean genuine understanding is happening at all — a critique worth returning to once this course reaches the era of large language models. Modern conversations about whether a chatbot "passes the Turing Test" happen constantly, generally without the nuance Turing's own paper actually contained.

🤔 Questions to Sit With

Reflection 1

Turing replaced "can machines think?" with "can a machine convincingly imitate a human in conversation?" Are these genuinely equivalent questions, or does something get lost in that substitution?

Reflection 2

Lady Lovelace's Objection and Turing's rebuttal to it are still argued about with modern machine learning systems in mind. Based on what you know of how modern AI systems are built, whose side of that 1950 argument do you find more convincing today?

Reflection 3

Turing anticipated and rebutted his critics directly inside his own paper — an unusually defensive structure for a piece of serious academic writing. What does that tell you about how controversial he expected this idea to be in 1950?

🎯 What's Next

Next chapter: **The Dartmouth Workshop (1956)** — the summer research project, directly inspired by thinkers like Turing, where the term "Artificial Intelligence" was formally coined and the field was born as an organized discipline.

🏫 The Dartmouth Workshop (1956)

Turing gave the field a testable question. This chapter covers the moment the field got a name, a founding document, and — despite lasting barely two months and being attended far more loosely than most retellings suggest — the event now almost universally cited as the formal birth of Artificial Intelligence as an academic discipline.

📝 The Proposal (1955)

In the summer of 1955, four researchers — **John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon** — submitted a funding proposal to the Rockefeller Foundation for a summer research project the following year at Dartmouth College. Its opening line is now one of the most quoted sentences in the field's history:

From the Proposal

> "We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College... The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."

McCarthy, Minsky, Rochester & Shannon, "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence" (1955)

This is the document that coined the term "artificial intelligence" itself. McCarthy chose the phrase deliberately — partly to distinguish this new proposed field from "cybernetics," an existing research program associated with Norbert Wiener that covered related but distinct ground (feedback and control in both machines and biological systems), and McCarthy wanted a name that wouldn't tie the new field to Wiener's specific framework or institutional territory.

🧑‍🤝‍🧑 What Actually Happened That Summer

Popular retellings sometimes imagine Dartmouth as a grand, formally structured conference. In reality, it was a loosely organized, informally attended summer workshop — participants came and went over roughly six to eight weeks, attendance at any given moment was far smaller than the proposal's "10 man" target, and there was no fixed daily agenda in the way a modern academic conference would have one.

Who Was There (at various points)

Beyond the four proposal authors, other attendees over the summer included **Allen Newell** and **Herbert Simon** (who would go on to be hugely influential in early AI and cognitive science), **Arthur Samuel** (known for pioneering machine learning work on checkers-playing programs), **Trenchard More**, **Ray Solomonoff**, and **Oliver Selfridge**. It was a small, informal gathering of a handful of researchers who would go on to define the field for decades — not a large, well-attended event by the standards of modern conferences.

🧠 The Logic Theorist — Arguably the First Real AI Program

Newell and Simon brought something genuinely remarkable to Dartmouth: the **Logic Theorist**, a program capable of proving mathematical theorems from Bertrand Russell and Alfred North Whitehead's *Principia Mathematica* — and, in at least one case, finding a proof more elegant than the one in the original text. It's widely regarded as the first program that could reasonably be called "artificial intelligence" in a substantive sense: not a fixed calculation, but a system searching through possibilities to find a valid logical proof, a genuinely different kind of computation than anything that had come before it.

⚠ The Seeds of Later Trouble

A Two-Month Timeline for "Every Aspect of Intelligence"

Read the proposal's opening line again: the founders genuinely believed a two-month summer project could make serious, foundational progress toward simulating "every aspect of learning or any other feature of intelligence." This wasn't unusual caution followed by pleasant surprise — it was the field's founding document setting an extraordinarily optimistic tone from day one. That same pattern — bold, confident predictions about how soon general machine intelligence would arrive — recurs constantly throughout the rest of this course, and directly sets up Chapter 5's First AI Winter, when funding bodies eventually ran out of patience with promises like this one going unfulfilled on schedule.

🏛️ Legacy

Despite falling enormously short of its proposal's stated ambitions within the actual two months, Dartmouth is near-universally cited as the formal founding event of AI as an organized academic field — the moment it got a name, a founding document, and a small core group of researchers who would go on to establish the field's major institutions. McCarthy later founded the Stanford AI Lab; Minsky co-founded the MIT AI Lab. The workshop didn't solve intelligence in a summer, but it did something arguably more durable: it created a community, a shared vocabulary, and an institutional foothold for the decades of research that followed.

🔮 Echoes We'll See Again

The gap between Dartmouth's founding optimism and what the field could actually deliver on that timeline is the first instance of a pattern that repeats at least twice more in this course: bold promises, genuine early progress, then a harder-than-expected wall, followed by disappointed funders pulling back. Keep this proposal's opening line in mind heading into Chapter 5.

🤔 Questions to Sit With

Reflection 1

The Dartmouth proposal's core conjecture — that intelligence can "in principle" be precisely described and simulated — is a philosophical claim, not just a technical one. Do you think that claim has actually been settled by anything covered so far in this course, or is it still an open question?

Reflection 2

McCarthy deliberately chose "artificial intelligence" over the already-established "cybernetics," partly for institutional/branding reasons. How much do you think the name a field gets called shapes how it's perceived and funded, independent of the actual research being done?

Reflection 3

The Logic Theorist could find genuinely novel, more elegant proofs than the textbook it was drawing from. Does that count as creativity, in your view — and if not, what would you want to see from a machine before you'd call it creative?

🎯 What's Next

Next chapter: **Early Programs** — the Logic Theorist's successor, the General Problem Solver, and the chatbot that fooled people into thinking it understood them: ELIZA.

💻 Early Programs

Dartmouth gave the field a name and a founding community. This chapter covers what that community actually built in the years right after — an ambitious universal problem-solver that revealed the field's first real limits, and a simple chatbot that became, by accident, one of the most important experiments in the entire history of human-computer interaction.

🧩 The General Problem Solver (1957)

Fresh from the Logic Theorist's success at Dartmouth (Chapter 2), Newell, Simon, and J.C. Shaw set an even more ambitious goal: a single program capable of solving *any* problem that could be formally represented — not just logic proofs, but puzzles, formal reasoning tasks, anything reducible to a clear starting state and a goal state. They called it the **General Problem Solver**, or GPS.

Means-Ends Analysis

GPS worked by comparing its current state to the desired goal state, identifying the biggest difference between them, and searching for an action that would reduce that difference — repeating the process until the gap closed entirely. It could genuinely solve problems like the Tower of Hanoi this way, treating wildly different puzzle types as instances of the same underlying search structure.

⚠ Impressive on Toy Problems, Stuck on Reality

GPS's fundamental limitation set the tone for an enormous amount of AI research to follow: it worked well on cleanly formalized problems with well-defined states and rules, and essentially couldn't function on anything resembling the genuine messiness of the real world — ambiguous goals, incomplete information, situations that resist being reduced to a tidy "current state vs. goal state" comparison. This gap — between dazzling performance on a clean, formal toy problem and near-total failure on anything genuinely real-world — recurs constantly throughout the rest of this course, right up to the deep learning era in Course 3.

🛋️ ELIZA (1966)

Joseph Weizenbaum, at MIT, built something far simpler than GPS in ambition, and far more consequential in what it accidentally revealed. **ELIZA** simulated conversation with a Rogerian psychotherapist — a real therapeutic style built around reflecting a patient's own statements back to them as open-ended questions, which turned out to be an almost perfect match for what simple text pattern-matching could actually pull off.

User: Men are all alike.
ELIZA: In what way?

User: They're always bugging us about something or other.
ELIZA: Can you think of a specific example?

User: Well, my boyfriend made me come here.
ELIZA: Your boyfriend made you come here?

Underneath, ELIZA had no model of meaning at all — it matched keywords against a script (Weizenbaum's most famous script was named **DOCTOR**), applied simple transformation rules (swapping "my" for "your," "I" for "you"), and reflected fragments of the user's own input back as a question, falling back to generic prompts when nothing matched. No understanding, no memory of the conversation beyond simple pattern rules, no actual model of what a "boyfriend" or "bugging" meant.

😮 The "ELIZA Effect"

Even Weizenbaum Was Unsettled

Weizenbaum reported being genuinely disturbed by how readily people attributed real understanding and empathy to ELIZA — most famously, his own secretary reportedly asked him to leave the room so she could have a private conversation with the program, despite knowing exactly how it worked and having watched it being built. This phenomenon was later named the **"ELIZA effect"**: the tendency to unconsciously treat a computer's behavior as though it reflects genuine human-like understanding, even when the observer knows perfectly well the underlying mechanism is simple and mechanistic.

A Real-World Confirmation of Course 1's Central Pattern

ELIZA is, in a very real sense, people unconsciously running their own private Turing Test (Chapter 1) — and the machine passing it, for many users, despite the mechanism being about as simple as a mechanism can get. This is the clearest confirmation yet, in this course's actual timeline, of the Mechanical Turk pattern from Course 1: a sufficiently well-targeted convincing performance gets believed, almost regardless of how sophisticated the thing producing it actually is.

🔄 From AI Success Story to AI's Sharpest Critic

What happened next is genuinely striking: Weizenbaum, the creator of one of the most famous early AI "successes," became one of the field's most prominent and outspoken skeptics as a direct result of what he'd witnessed. His 1976 book *Computer Power and Human Reason* argued forcefully against trusting computers with tasks requiring genuine human judgment, wisdom, or compassion — precisely because he'd seen firsthand how easily people could be fooled into over-trusting a system with no genuine understanding at all.

📜 Two Programs, Two Very Different Legacies

	General Problem Solver (1957)	ELIZA (1966)
Goal	Solve any formally-representable problem	Simulate a specific style of conversation
Underlying approach	Means-ends analysis, state-space search	Keyword pattern-matching and reflection
Where it succeeded	Clean, formal toy problems (Tower of Hanoi)	Convincing many real users of genuine understanding
Lasting legacy	Exposed the toy-problem-vs-reality gap	Named a whole psychological phenomenon (the ELIZA effect)

🤔 Questions to Sit With

Reflection 1

ELIZA's specific choice of a Rogerian therapist persona (which naturally involves reflecting statements back as questions) was arguably a big part of why it worked as well as it did. Do you think ELIZA would have been nearly as convincing simulating a different kind of conversation — say, a friend giving advice?

Reflection 2

Weizenbaum's secretary knew exactly how ELIZA worked and still wanted a private conversation with it. Have you ever caught yourself experiencing something like the ELIZA effect — responding emotionally to a system you know, intellectually, has no real understanding?

Reflection 3

GPS aimed to be maximally general and struggled everywhere except clean formal problems; ELIZA aimed to be narrowly specific and accidentally fooled real people. What does the contrast between these two outcomes suggest about the relationship between a system's scope of ambition and how convincing it actually turns out to be?

🎯 What's Next

Next chapter: **Symbolic AI / GOFAI** — the broader "Good Old-Fashioned AI" paradigm that Logic Theorist, GPS, and most of this era's research shared, and the core assumption behind it that would eventually prove to be its greatest weakness.

🔣 Symbolic AI / GOFAI

Logic Theorist, GPS, and ELIZA look quite different on the surface — a theorem prover, a puzzle solver, a chatbot. This chapter steps back to name the single underlying paradigm all three actually shared, the confident philosophical claim behind it, and the specific weakness in that claim that would eventually define the rest of this course's Real AI era.

🏷️ A Name Coined in Hindsight

The term "GOFAI" — **Good Old-Fashioned AI** — wasn't used by the researchers building this era's programs at the time. Philosopher John Haugeland coined it in 1985, retrospectively, specifically to distinguish this earlier symbol-manipulation approach from the different techniques (including the statistical, learning-based methods covered starting in Course 3) that were beginning to compete with it by the mid-1980s. Naming a paradigm "old-fashioned" is itself a signal that something newer had already started to take its place.

🧠 The Physical Symbol System Hypothesis

Newell and Simon — already met twice in this course, for the Logic Theorist and GPS — formalized the core assumption behind the entire era in a 1976 paper, as the **physical symbol system hypothesis**:

The Claim

> "A physical symbol system has the necessary and sufficient means for general intelligent action."

In plain terms: a system that manipulates discrete symbols according to formal rules — nothing more, nothing categorically different — is both *enough* to produce genuine intelligence, and *required* for it. Not one possible route to intelligence among several; the necessary foundation for any intelligence at all, artificial or otherwise.

This is a genuinely strong philosophical claim, and it's worth being clear that it was, for two decades, close to the field's unquestioned default assumption — not a hedge, not one hypothesis among competing schools of thought, but the working foundation nearly everything covered so far in this course was built on top of.

🧊 SHRDLU (1970) — GOFAI's High-Water Mark

Terry Winograd's **SHRDLU**, built at MIT, is frequently cited as the clearest and most impressive demonstration of what symbolic AI could achieve. SHRDLU could hold a genuine conversation about a simulated "blocks world" — colored blocks, pyramids, and boxes on a virtual tabletop — understanding and correctly executing instructions like "find a block which is taller than the one you are holding and put it into the box," tracking pronoun references across a conversation, and even explaining its own past actions when asked why it had done something.

Impressive — Within an Extremely Narrow World

SHRDLU's apparent understanding worked because its entire universe of discourse — every object, every possible relationship, every valid action — was small, closed, and fully specified in advance. Language understanding, logical reasoning about the block world's state, and action planning were all tightly integrated, which is exactly why it worked so well: there was no open-ended real-world ambiguity for the symbolic representation to fail to capture.

⚠ Where the Hypothesis Started to Break

The Knowledge Acquisition Bottleneck

Every fact, every rule, every piece of world knowledge a symbolic system uses has to be manually identified and hand-encoded by a human — there's no mechanism for the system to learn new knowledge from raw experience the way Course 3's statistical approaches eventually would. This became known as the **knowledge acquisition bottleneck**: human common-sense knowledge is vast, constantly context-dependent, and full of exceptions, and hand-encoding even a meaningful fraction of it turned out to be a staggering undertaking. The **Cyc project**, begun in 1984 with the explicit goal of hand-encoding millions of pieces of everyday common-sense knowledge as formal symbolic rules, is *still ongoing today* — a genuinely sobering illustration of just how large the task actually is, decades after it began.

Closely related: symbolic systems tend to be **brittle** — SHRDLU inside its blocks world was remarkably capable, but the same architecture couldn't be meaningfully extended to, say, understanding a conversation about cooking, or sports, or anything outside its narrowly pre-specified domain, without essentially starting the hand-encoding process over from scratch. A system built entirely on explicit rules tends to fail completely, often nonsensically, the moment it steps outside the boundaries someone thought to encode in advance.

📜 Strengths and Weaknesses, Side by Side

Strength	Corresponding Weakness
Reasoning is transparent — you can read the exact rule that produced a conclusion	Every rule has to be written by hand, one at a time
Excellent for closed, well-defined domains (theorem proving, SHRDLU's blocks world)	Brittle and often useless the moment the domain's boundary is crossed
Grounded in a clear, principled philosophical hypothesis	The hypothesis itself doesn't scale to the genuine scope of real-world common sense

🔮 The Pivot That's Coming

The knowledge acquisition bottleneck is the single biggest reason the field eventually shifted away from hand-encoded symbols toward the statistical, learn-from-data approach that dominates modern AI — the entire subject of Course 3. Instead of a person writing down every rule in advance, a statistical system infers patterns directly from large amounts of real-world data. That pivot doesn't happen for decades yet in this course's timeline — but the specific problem driving it is fully visible right here, in the 1970s and 80s, well before anyone had the data or computing power to actually attempt the alternative.

🤔 Questions to Sit With

Reflection 1

The physical symbol system hypothesis claims symbol manipulation is both necessary and sufficient for intelligence. Based on everything covered in this course so far, do you find the "necessary" half or the "sufficient" half of that claim more convincing — or less?

Reflection 2

The Cyc project has been hand-encoding common-sense knowledge since 1984 and is still not finished. What does that timeline tell you about the actual scale of "common sense," compared to how casually the phrase gets used in everyday conversation?

Reflection 3

SHRDLU could explain its own reasoning in a way that's genuinely harder for many modern statistical AI systems to do. Is transparent, explainable reasoning worth trading away for broader real-world capability — or should both be achievable at once?

🎯 What's Next

Next chapter: **The First AI Winter** — the Lighthill Report, the funding collapse that followed, and how the knowledge acquisition bottleneck this chapter covered helped bring it about.

❄️ The First AI Winter

Chapter 4 ended on a warning: the knowledge acquisition bottleneck and the brittleness it caused weren't just an academic footnote — they were about to collide with real funding decisions. This chapter covers what happens when a field's founding promises (Chapter 2's confident two-month proposal) go unfulfilled for long enough that the people paying for it stop believing.

🌐 Machine Translation Goes First: The ALPAC Report (1966)

Machine translation was one of AI's earliest and most heavily funded practical applications, driven substantially by Cold War interest in automatically translating Russian scientific and military documents. By the mid-1960s, progress had badly stalled — translations were frequently unusable, and the U.S. government commissioned the **Automatic Language Processing Advisory Committee (ALPAC)** to assess whether continued investment was justified.

The 1966 Verdict

ALPAC's report found that machine translation had made little genuine progress, was slower and less accurate than human translation, and did not justify its funding level. This was the **first** major funding blow to AI-adjacent research — arriving years before the more famous Lighthill Report, and specifically targeting one narrow application rather than the field as a whole, but establishing the pattern the rest of this chapter follows.

GB The Lighthill Report (1973)

In 1973, the UK's Science Research Council commissioned mathematician **Sir James Lighthill** — notably, not an AI researcher himself — to assess the state of AI research and whether continued government funding was justified.

A Harsh Verdict, and a Specific Technical Criticism

Lighthill's report concluded that AI's grand promises had gone substantially unfulfilled, and singled out a specific technical failure as the central problem: the field's inability to deal with **combinatorial explosion** — the way the number of possibilities a symbolic search algorithm has to consider grows exponentially as a problem's real-world complexity increases, quickly outpacing any realistic amount of computing power. GPS (Chapter 3) and most of the era's symbolic search techniques (Chapter 4) worked precisely because their problems were kept small and clean; Lighthill's report argued the field had failed to show any credible path to scaling those same techniques up to genuinely complex, real-world problems.

The report recommended sharply reducing AI funding in the UK — and it landed. Several UK university AI research groups were severely cut back or shut down entirely in the years that followed.

📺 The Lighthill Debate

In a genuinely unusual public confrontation for a scientific dispute, the BBC broadcast a televised debate over the report's conclusions, pitting Lighthill against defenders of the field including **Donald Michie, John McCarthy** (Chapter 2's Dartmouth co-organizer), and psychologist Richard Gregory. The debate didn't reverse the funding decisions, but it stands as a rare, vivid moment where an entire research field's legitimacy was argued out in front of a general television audience rather than settled quietly within academic journals.

US A Chilling Effect Beyond the UK

Though the Lighthill Report was a UK-specific document, its conclusions — and ALPAC's earlier verdict — contributed to a broader loss of confidence in AI research funding across the US as well, even as some pockets of funding (notably DARPA-backed speech recognition work) continued in narrower, more defensible areas. The pattern that emerged — a wave of enthusiastic funding driven by bold promises, followed by a sharp pullback once those promises weren't met on schedule — was later given a name, coined by analogy to "nuclear winter": **AI winter**.

📜 Cause and Effect

Root Cause (Chapter 4)	Consequence (This Chapter)
Knowledge acquisition bottleneck — hand-encoding doesn't scale	Machine translation stalls; ALPAC recommends cuts (1966)
Combinatorial explosion — symbolic search doesn't scale to real problems	Lighthill Report recommends severe UK funding cuts (1973)
Founding promises (Chapter 2's Dartmouth proposal) go unfulfilled on schedule	Funders lose patience; the field's credibility takes a lasting hit

🔮 Not Bad Luck — a Direct Consequence

It's worth being precise about the causal chain here: this wasn't an unrelated funding accident that happened to hit AI research at a bad moment. The Lighthill Report's central technical criticism — combinatorial explosion — is a direct, specific consequence of the exact symbolic-search limitations covered in Chapter 4. The field's founding optimism (Chapter 2) collided with a genuine, well-documented technical wall, and the funding collapse followed as a fairly predictable result.

🤔 Questions to Sit With

Reflection 1

Lighthill was a mathematician, not an AI researcher, assessing a field he wasn't a specialist in. Does that make his report's conclusions more or less credible — an outsider's clear-eyed view, or a lack of genuine domain expertise?

Reflection 2

A televised public debate over a research field's legitimacy is genuinely unusual. Can you think of a more recent scientific or technological field that's faced something comparable — a public, televised or widely broadcast argument over whether it deserves continued funding or trust?

Reflection 3

The Dartmouth proposal's two-month timeline (Chapter 2) and the Lighthill Report's harsh correction seventeen years later are two ends of the same boom-bust cycle. Whose responsibility do you think it is to prevent this pattern — the researchers making promises, the funders believing them, or is it simply an unavoidable feature of funding ambitious, uncertain research?

🎯 What's Next

Next chapter: **Expert Systems** — how AI research found its way back to significant funding and commercial success in the 1980s by abandoning general intelligence ambitions in favor of narrow, practical domains like medical diagnosis.

Expert Systems

Chapter 5 ended in a funding collapse driven by a specific, well-documented failure: symbolic AI's inability to hand-encode the vast, messy scope of real-world common-sense knowledge. This chapter covers how the field found its way back to funding and genuine commercial success in the 1980s — not by solving that problem, but by sidestepping it, narrowing ambition from "general intelligence" down to "one expert's specialized knowledge in one narrow domain."

🔧 What Is an Expert System?

An **expert system** is a program built from two separable pieces: a **knowledge base** of hand-coded if-then rules capturing a human expert's domain-specific know-how, and an **inference engine** — a general-purpose piece of software that applies those rules to a specific situation to reach a conclusion, independent of what the rules actually say. Separating the two mattered enormously: the same inference engine could be reused across completely different domains just by swapping in a new rule set, which is exactly what let the "expert system shell" industry (later in this chapter) exist as a business at all.

🧪 DENDRAL (1965) — The First True Expert System

Built at Stanford starting in 1965 by **Edward Feigenbaum**, **Bruce Buchanan**, and Nobel laureate geneticist **Joshua Lederberg**, DENDRAL inferred the molecular structure of organic compounds from raw mass spectrometry data, using hand-encoded chemistry rules to narrow down which molecular structures could plausibly produce a given spectrum.

Feigenbaum's Central Lesson

DENDRAL's success led Feigenbaum to a philosophy he called the **"knowledge is power"** hypothesis: intelligent behavior comes primarily from having a large amount of specific, narrow domain knowledge — not from a single clever general reasoning method. This is a direct, quiet reply to Chapter 4's physical symbol system hypothesis. It doesn't reject symbol manipulation; it reframes what actually does the work. Not a more powerful general reasoner, but more — and narrower — facts.

💊 MYCIN (1972) — Diagnosing Blood Infections

Led by **Ted Shortliffe** at Stanford, MYCIN used roughly **600 hand-coded rules** to diagnose bacterial blood infections (such as bacteremia and meningitis) and recommend antibiotic treatment, with dosages adjusted for the patient's weight. To handle medicine's inherent uncertainty, MYCIN attached a **certainty factor** — a numeric confidence score — to each rule and conclusion, an early, pragmatic alternative to full probability theory.

The System That Worked, But Was Never Used

In blind evaluation studies, MYCIN's treatment recommendations were judged acceptable more often than those of junior doctors, and were competitive with infectious-disease specialists. And yet MYCIN was **never adopted for actual clinical use**. The obstacles weren't technical: unresolved liability (who is responsible if a computer's medical advice turns out wrong?), the difficulty of fitting a rule-based advisor into real hospital workflow, and the ongoing effort required to keep 600 rules current as medical knowledge evolves. A capable system and a deployed system turned out to be two very different achievements.

💰 XCON / R1 (1980) — The Commercial Proof Point

Built by **John McDermott** at Carnegie Mellon and deployed by **Digital Equipment Corporation (DEC)**, XCON (also called R1) automatically configured customer orders for DEC's VAX minicomputers — translating a sales order into the complete, correct list of components, catching errors human sales engineers routinely made. Unlike DENDRAL and MYCIN, XCON wasn't a research demonstration; it was a real production system with a real dollar figure attached.

By the Numbers

By 1986, XCON was processing roughly **80,000 orders a year** and was credited with saving DEC on the order of **$40 million annually**. This was the concrete, widely publicized evidence the field had been missing since Chapter 5's funding collapse: proof that a symbolic, rule-based system could deliver direct, measurable business value — not someday, but right now, in production.

🌎 The Boom, and Its International Reach

XCON's visible success — alongside DENDRAL and MYCIN's academic credibility — triggered a genuine commercial gold rush. Startups like **Teknowledge** and **Intellicorp** sold "expert system shells": the inference-engine half of the architecture, packaged as a reusable product a company could plug its own domain rules into, without building an inference engine from scratch. Because LISP was AI's dominant programming language, companies including **Symbolics** and **Lisp Machines Inc.** began selling specialized workstation hardware — **LISP machines** — built specifically to run it fast.

The boom wasn't confined to the US. In 1982, Japan launched the government-funded **Fifth Generation Computer Systems** project, an ambitious state effort explicitly aimed at leapfrogging the West in AI and logic-programming hardware. That move spurred competitive funding responses abroad — DARPA's Strategic Computing Initiative in the US, the Alvey Programme in the UK — turning expert systems into something closer to a national-competitiveness race than a purely academic pursuit.

📜 Narrowing the Ambition

Symbolic AI's General Ambition (Chapter 4)	Expert Systems' Narrow Focus (This Chapter)
Aimed at general intelligent action — the physical symbol system hypothesis	Deliberately abandoned generality, targeting one professional domain at a time
Knowledge acquisition bottleneck was fatal at "common sense" scale	The bottleneck became manageable once the knowledge needed was just one expert's few hundred rules
Brittle the moment SHRDLU's blocks-world boundary was crossed	Still just as brittle outside its domain — but the domain (VAX orders, blood infections) was valuable enough to justify the narrowing

🔮 A Partial Answer, Not a Full One

It's worth being precise about what actually happened here: the knowledge acquisition bottleneck and brittleness that ended Chapter 5 weren't solved — they were sidestepped, by shrinking the target until hand-encoding became feasible again. That's a genuinely useful engineering trick, and it built real, profitable systems. But it also means the entire 1980s boom — the shells, the LISP machines, the national funding races — was built on the same fundamentally brittle foundation as SHRDLU, just aimed at smaller targets. What happens when a hardware and software industry has been built entirely around betting that this narrowing trick keeps paying off is exactly where the next chapter picks up.

🤔 Questions to Sit With

Reflection 1

Feigenbaum's "knowledge is power" hypothesis and Chapter 4's physical symbol system hypothesis aren't opposites — but they emphasize very different things. Does narrowing a domain until hand-encoding becomes feasible actually address the knowledge acquisition bottleneck, or does it just relocate the boundary where the system breaks?

Reflection 2

MYCIN outperformed junior doctors in blind testing but was never used clinically, largely over liability and workflow concerns rather than raw competence. Does that gap between "tested capability" and "real-world deployment" sound familiar from anything happening with AI systems today?

Reflection 3

Japan's Fifth Generation project triggered competitive funding responses in the US and UK, echoing the Cold War funding dynamics that drove machine translation research in Chapter 5. Is a competitive, race-driven funding environment more likely to produce durable progress, or more likely to inflate a bubble that eventually bursts?

🎯 What's Next

Next chapter: **The Second AI Winter** — how the very commercial infrastructure this chapter describes, the expert system shells and the specialized LISP machines built to run them, collapsed almost as fast as it rose.

🥶 The Second AI Winter

Chapter 6 closed with a warning: the entire 1980s expert systems boom — the shells, the specialized LISP hardware, the national funding races — was built on the same brittle, hand-encoded foundation as SHRDLU, just aimed at smaller targets. This chapter covers what happened when that bet stopped paying off, and why this second collapse looked less like a research reckoning and more like an ordinary market crash.

🖥️ The Lisp Machine Crash

Companies like **Symbolics** and **Lisp Machines Inc. (LMI)** had built entire businesses around specialized workstations engineered to run LISP fast — hardware tuned for exactly the language the expert-systems boom depended on. For a few years in the early 1980s, that specialization was a genuine performance advantage.

It didn't last. General-purpose workstations from companies like **Sun Microsystems** and **Apollo**, built on conventional processors, improved rapidly — and paired with better LISP compilers, they closed the performance gap while costing a fraction of the price. By the mid-1980s, buying a purpose-built LISP machine no longer made economic sense when a cheaper general-purpose workstation could run the same software nearly as well.

By the Numbers

Symbolics stock peaked in **1987** and then collapsed almost immediately as the specialized-hardware market it depended on evaporated. Across the LISP machine industry, companies that had raised significant investment on the strength of the expert-systems boom were bankrupt, acquired, or wound down within just a few years — an abrupt, market-driven crash rather than a slow fade.

⚙️ Why Expert Systems Stopped Scaling

The narrowing trick that revived the field in Chapter 6 had a hidden cost that only showed up once systems grew large. A handful of rules is easy to reason about; thousands of rules start interacting with each other in ways no one predicted when writing any single rule in isolation. Adding a new rule to fix one case could silently break the reasoning behind an unrelated one.

XCON's Own Growing Pains

Even XCON — the very system Chapter 6 held up as proof that expert systems delivered real commercial value — wasn't immune. Its rule base reportedly grew from a few thousand rules to somewhere around ten thousand over the following decade, and keeping it consistent and correct became a substantial, ongoing engineering effort in its own right. Many companies that built smaller in-house expert systems found the long-term maintenance cost eventually outweighed the benefit, and quietly abandoned them.

📢 A Term Coined Before the Crash

Here's a detail worth sitting with: the phrase **"AI winter"** itself — introduced back in Chapter 5 to describe the Lighthill-era collapse — wasn't actually coined until **1984**, at an AAAI (American Association for Artificial Intelligence) conference panel featuring **Marvin Minsky** and **Roger Schank**. They used it as a warning about an approaching pullback they saw coming in the then-current expert-systems hype. The term predates the very winter this chapter describes — a rare case of researchers naming a boom-bust pattern accurately enough to predict its next occurrence a few years in advance.

💸 DARPA Pulls Back

Government funding followed the same trajectory as private investment. DARPA's Strategic Computing Initiative, launched in the early 1980s partly in response to Japan's Fifth Generation project (Chapter 6), had promised ambitious near-term military and industrial applications of AI. When those promises didn't materialize on schedule, funding was sharply reduced in the late 1980s — echoing, on a smaller scale, the exact funding-versus-promises dynamic that produced the Lighthill Report in Chapter 5.

📜 Two Winters, Compared

First AI Winter (Chapter 5)	Second AI Winter (This Chapter)
A research-community reckoning — the Lighthill Report assessed unmet promises	A market correction — commercial hype met real maintenance costs and cheaper competing hardware
Hit government-funded academic research directly	Hit a private industry built on venture capital and corporate contracts
Triggered by a technical limit — combinatorial explosion in symbolic search	Triggered by a practical limit at scale — rule interaction complexity — plus being economically outcompeted

🔮 The Pattern Repeats On Schedule

Two winters now, separated by roughly a decade, each following the same underlying shape: bold promises attract funding and investment, the field delivers real but narrower results than promised, and the gap between promise and delivery eventually triggers a collapse. Minsky and Schank were able to name the pattern accurately enough to warn about it in advance. The next chapter asks the harder question this course has been building toward: what exactly is the repeating mechanism behind that pattern, and does recognizing it in advance actually help anyone avoid it?

🤔 Questions to Sit With

Reflection 1

The Lisp machine crash was primarily an economic story — cheaper general-purpose hardware caught up — rather than a story about AI research failing on its own terms. Does that make this winter feel less serious than the first one, or does a purely market-driven collapse carry its own kind of warning?

Reflection 2

Minsky and Schank named "AI winter" as a warning in 1984, and the winter they warned about arrived a few years later anyway. What does it tell you that accurately predicting a hype cycle didn't seem to prevent it?

Reflection 3

Rule-based systems became hard to maintain once they scaled past a few thousand rules, because new rules could interact unpredictably with old ones. Can you think of other kinds of complex systems — software or otherwise — where growth itself, rather than any single flaw, becomes the main source of fragility?

🎯 What's Next

Next chapter: **Lessons From Two Winters** — the final chapter of this course, naming the recurring hype-cycle pattern directly and bridging to Course 3's statistical, data-driven era.

🌉 Lessons From Two Winters

Chapter 7 ended on a hard question: Minsky and Schank named the "AI winter" pattern accurately enough to warn about it in advance, and it happened anyway. This final chapter of Course 2 names the mechanism behind that pattern directly, traces it across everything covered so far, and asks what — if anything — actually breaks it, as a bridge into Course 3's very different technical era.

🔄 The Hype Cycle, Named

Strip away the specific technologies and both winters covered in this course follow the identical shape: a bold, specific promise attracts funding or investment; real work delivers genuine but narrower results than promised; the gap between promise and delivery eventually becomes impossible to ignore; funding or investment collapses sharply. This isn't unique to AI — the technology analyst firm Gartner later formalized a general version of this shape as the **"Hype Cycle"** in 1995, and AI's own boom-bust history is frequently cited as one of the clearest, earliest examples of the pattern it describes.

📜 The Whole Course, at a Glance

Chapter	Event	Pattern That Recurs
1	Turing & the Imitation Game	A serious, testable question replaces centuries of fictional speculation (Course 1)
2	The Dartmouth Workshop (1956)	A confident, specific promise — general AI in one summer — sets the bar impossibly high
3	Logic Theorist, GPS, ELIZA	Genuine early wins, achieved only inside small, tightly bounded problems
4	Symbolic AI / GOFAI	A strong general hypothesis meets the knowledge acquisition bottleneck
5	The First AI Winter	Unmet founding promises collide with funders' patience — the first collapse
6	Expert Systems	Recovery by narrowing scope, not by solving the underlying bottleneck
7	The Second AI Winter	The narrowed bet fails too, once at commercial scale — the second collapse

🎯 Why the Pattern Kept Repeating

It's tempting to treat each winter as a one-off mistake — a bad report, an overhyped demo, a market miscalculation. Looking at both side by side suggests something more structural. Researchers need funding to continue their work, and confident, ambitious claims are what attracts it; funders and the public, in turn, tend to hear a narrow technical demonstration (SHRDLU's blocks world, XCON's order forms) and extrapolate it into a much broader claim about "real" intelligence than the researchers themselves may have intended. Neither side is straightforwardly lying — but the incentive to overstate, and the tendency to over-hear, point in the same direction every time.

Recovery Without a Fix

Notice what Chapter 6's recovery actually did, and didn't do. Expert systems didn't solve the knowledge acquisition bottleneck that ended Chapter 5 — they made it manageable by shrinking the target until a human could realistically hand-encode it. That's a real, useful engineering move, and it built genuinely profitable systems. But it left the fundamental limitation untouched, which is precisely why Chapter 7's collapse — at a larger scale, on a longer fuse — was, in hindsight, close to inevitable rather than an unrelated market accident.

🧿 What Actually Breaks the Cycle

Both winters in this course share a single root cause: symbolic AI's absolute dependence on a human being manually encoding every fact and rule the system would ever use. Course 3 opens with a genuinely different approach — systems that learn patterns directly from large amounts of real-world data, rather than requiring a person to write every rule down in advance. That doesn't automatically mean the hype cycle disappears (Course 3's own chapters will show it hasn't, entirely) — but it does mean the *specific* bottleneck that caused both of this course's winters no longer applies in the same way. Removing a root cause, rather than narrowing around it, is the difference this course's whole arc has been building toward.

🤔 Questions to Sit With

Reflection 1

Both winters trace back to the same root cause — hand-encoded knowledge doesn't scale — approached from two different angles. If Course 3's data-driven approach removes that specific bottleneck, do you expect the boom-bust hype cycle to disappear entirely, or just to find a new bottleneck to break on?

Reflection 2

This chapter suggests neither researchers nor funders are simply "at fault" — overstating and over-hearing reinforce each other. Whose responsibility do you think it actually is to keep that dynamic in check: the researchers, the funders and media, or does it require some kind of outside structure neither side can be trusted to provide alone?

Reflection 3

Gartner's Hype Cycle was formalized for technology broadly, not just AI. Thinking back over other major tech shifts you've lived through or read about, does this same overpromise-collapse-recover shape show up outside computing too, or is there something specific to AI that makes it especially prone to this pattern?

🎯 What's Next

Course 2 (The Birth & Growth of Real AI) is complete. Course 3 (The Statistical & Deep Learning Era) begins with **The Statistical Turn** — how machine learning's data-driven approach overtook symbolic AI, and why it sidesteps the exact bottleneck that caused both winters in this course.