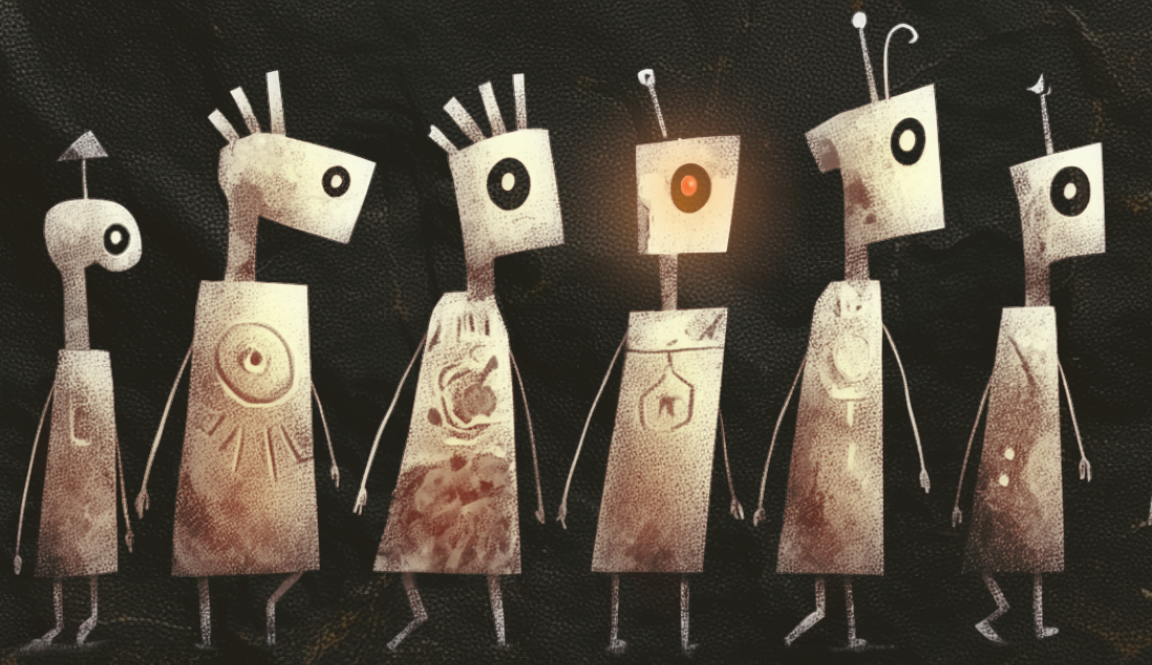
INGA STRÜMKE

MACHINES THAT THINK

HOW ARTIFICIAL INTELLIGENCE WORKS AND WHAT IT MEANS FOR US



Rheinwerk Publishing

Contents

Foreword 7
Preface 11

Part I: The Start of It All

1. A Carefully Selected History 17
The World's Most Successful Hoax 17
Algorithms 19
Analog Computers 20
Gödel, Turing, and von Neumann 21
A Field of Study Is Forged 25
Checkmate 26
The Drosophila of Reasoning 31
The Best Chess Player in the Universe 32
The Age of Machines 35

2. The Attempt to Make Machines Intelligent 39
Symbolic AI 39
Expert Systems 40
The ELIZA Effect 46

Winter and Spring ... 49
Machines That Learn ... 51
Supervised Learning ... 53
Flowers and Decision Trees ... 56
Subsymbolic AI ... 59
Neural Networks Through the Ages ... 63
Learning Anything ... 67

3. The Hunt for Data ... 73
The True Distribution ... 73
Data for Machine Learning ... 76
The Wrong Distribution ... 78
Bias for Statisticians and Journalists ... 80
Data Is Expensive ... 85
Long, Problematic Tails ... 87
Finding Your Own Data ... 89

Part II: Artificial Intelligence Today

4. Seeing Is Believing ... 97
A World of Matrices ... 97
Machines That Almost Understand What They're Looking At ... 102
The Simple Answer Is Often Wrong ... 107
Machines That Understand More Than We Do ... 110
Artificial Intuition and Chess ... 114
Mental Health ... 117
Privacy ... 120
The Right to an Explanation ... 124

5. Is Anybody Home? ... 127
Artificial Conversation Partners ... 127
The Ghost in the Machine ... 130
ChatGPT and Concerned Teachers ... 135
Talk About Problems ... 140
A Turing Test for Consciousness ... 142

The Art of Creation 146
From Noise to Reality 153

6. Our Artificially Intelligent Life 161
The Machine Revolution 161
The World of Ethics 166
Someone Has to Decide 172
The War of the Currents and the AI Act 176
The Collective Action Problem of AI Development 181
Privacy in Flux 184

7. Attacking Machine Learning Models 191
Well-Behaved Machines with Good Intentions 191
Optical Illusions for Machines 194
Cutting Grass in Three Dimensions 198
The Curse of Dimensionality 200
The Back Way In 203
Winter and Politics 207
Autonomy and Control 213

Part III: Artificial Intelligence Tomorrow

8. Superintelligence and Other Speculation 221
Artificial Creativity 221
Speech and Acoustics 224
Data Use 226
AI Ethics and Moral Agents 228
Artificial General Intelligence and Superintelligence 231
Machine Goals, Human Goals, and Mesa-Optimization 238
A Brain Thinking About Itself 245

Illustrations 255

Preface

One late evening in the spring of 2018, 49-year-old Elaine Herzberg was pushing her bicycle across an Arizona highway. Her bike laden with shopping bags, she cut straight across the wide, four-lane road instead of using a crosswalk. Coming along the road toward Elaine was a Volvo XC90, but she kept walking—and the car didn't stop. It didn't even brake, instead hitting Elaine, who later died in the hospital from her injuries. It's remarkable that the car didn't brake—especially considering that this particular vehicle had two drivers: one human and one machine. The machine was controlling the car as part of Uber's self-driving test program, while the human—Rafaela Vasquez—was responsible for sitting behind the wheel and assuming control if anything went wrong. Both drivers failed that night, and it's easy to see why, at least in Rafaela's case. The vast majority of people become complacent when carrying out routine tasks such as overseeing machines that more or less do what they're supposed to. Footage from inside the car showed Rafaela looking down and smiling shortly before the collision, suggesting that she was focused on something other than the road. Still, Rafaela must have realized something was wrong at the last moment because she grabbed the steering wheel less than a second before the vehicle struck Elaine. She also applied the brakes shortly after impact, but by then, it was too late. The following year, in the spring of 2019, the county attorney declined to press charges

against Uber, while Rafaela was indicted by a grand jury on a count of negligent homicide.

Was it fair that Rafaela was blamed for the machine's mistake? In fact, declaring that the machine made a mistake isn't all that clear cut—it's far less apparent what went wrong on its side. With hundreds of thousands of miles already logged, this incident was the first time a human died after being struck by one of Uber's self-driving cars. This rarity meant that both Uber and Rafaela had good reason to assume that nothing would go wrong.

Before self-driving cars are put to use, they go through countless hours of training, both in simulated conditions and in real traffic. But if a human went through a similar training regime, we wouldn't assume that they would never make a mistake while out on the road: Humans are responsible for more than 6 million yearly car accidents just in the US; globally, 1.2 million people die in traffic accidents every single year. That's many times the number who die due to wars. The potential for improvement is striking, and if we look at the main causes of accidents—speeding, intoxication, and inattention—it's only natural to think that machines must be better at driving than us humans. Machines stick to speed limits; they don't drink or do drugs; and as long as they have power, they pay attention. Machines have reaction times and calculation capacities that we humans can only dream of. Nevertheless, machines make mistakes that humans never would. No one on the planet would have seen a pedestrian crossing the road and thought there was no need to brake.

So, what caused a machine to run Elaine over in 2018? The simple answer is that it classified her as a *false negative*. The machine observed her, but it incorrectly classified her as something that you don't stop for, like newspapers or plastic bags. Classification is among the most common tasks we currently use artificial intelligence for. What happens when that classification goes wrong and the error impacts people? "The machine got it wrong" is too simple an explanation, and it obscures where the responsibility really lies. A machine tasked with classifying objects that it sees while out on the road—pedestrians, cyclists, cars, motorcycles, trucks, and plastic bags—is developed using data that a human provided. As such, a machine never says things in terms of "human" or "plastic bag." The only thing machines output are numbers. So, once again it's the responsibility

of humans to determine the threshold value to classify what the machine is seeing as a human, a plastic bag, or something else entirely. Every time artificial intelligence is used in decision-making, there is always at least one tradeoff. And determining which tradeoffs are acceptable and which are not is the domain of humans. An engineer set the threshold that would have made the car stop—or in Elaine's tragic case, not.

It's the same story whenever artificial intelligence is involved: When we read about machines that discriminate, it's not because the machine itself is racist or sexist. The real issue lies in the tradeoffs made (or not made) during data collection and training. Artificial intelligence can solve problems, but we humans must ensure that machines end up solving the right problems. Those tradeoffs, and thus the responsibility for them, always rests with humans.

How can we know when it's safe to hand a task over to a machine? And when a machine fails, who should take the blame? Why risk letting machines loose in the world if they might make mistakes—and why create machines in the first place? Why are machines sometimes hundreds of times smarter than us, yet complete idiots at other times? Why do we humans have such a deep desire to build intelligent machines? Join me as we dive into all of these questions and many more.

Chapter 1
A Carefully Selected History

The World's Most Successful Hoax

Archduchess Maria Theresa of Austria must have been quite the lady. To impress her, the inventor Wolfgang von Kempelen built what would later be known as the world's most successful hoax: the Mechanical Turk. This marvelous machine led to the first headlines about machines outperforming humans at chess, as early as the '70s—the 1770s, that is. "The Turk" consisted of a mechanical man dressed in a cloak and turban, sitting on a box containing a jumble of cog wheels and seemingly complex machinery. It—he?—would begin a chess game by moving his head from side to side, as if looking at the chessboard and contemplating his first move. His arm would then suddenly shoot forward, and the mechanical fingers would pick up a piece, placing it on a legal square. This feat alone was not enough to impress the citizens of the 18th century, who had seen mechanical animals with far more convincing details than this crude imitation of a man. The truly groundbreaking aspect was the game itself: The Turk was *good* at chess. Really good. Even Napoleon Bonaparte himself was beaten by the Turk and went to his grave thinking that a machine had beaten him in chess. The Turk played creatively and adapted to his opponent's style, game after game. And if someone tried to cheat—as Napoleon did when he challenged the Turk in 1809—the Turk would simply move the piece back to its original

position. If an opponent repeatedly tried to cheat, the Turk would eventually lose his patience and swipe his arm across the board, sending the pieces flying.

Clearly, this was all just fraud, albeit in the shape of an engineering marvel. The intricate box containing numerous gears and cogs could be opened for inspection. As the audience peered inside and inspected it from all angles before each game, they were convinced that there was no human in the box. Yet speculation was plentiful, including one claim that a Russian soldier with a talent for chess—who had lost both his legs in combat—was the real brain behind, or more precisely inside, the Turk. When the hoax was finally revealed in *The Chess Monthly* in the late 1850s, it was said that "no secret was ever kept as the Turk's has been." The trick was that the person—who was very much inside the box—sat on a chair that could be moved around, allowing them to remain out of sight during inspections.

There are likely several reasons why a chess-playing machine would draw so much attention, and among these was the intriguing idea that a machine could be smarter than humans, that is, could make better *decisions* than humans. And although the world would still wait quite some time before a machine that could truly challenge humans at chess, by the 19th century, machines were already starting to change people's lives and reshape the entire social economy. The steam engine has become an iconic 18th-century invention, providing power on a scale unmatched by any animal, human, or technology of the time. The shift from human craftsmanship to mechanical production began in England, spread rapidly throughout Europe and to the United States, and became an industrial turning point. Rather than goods being produced by humans, machines were built to produce them more efficiently, and agriculture-based economies abruptly shifted to industrial, machine-based ones. In the Western world, where most people were farmers living off the land, this development caused a complete disruption to their way of life. And the Mechanical Turk arrived right in the middle of this era. It's no wonder that the idea of a machine capable of, not just working, but also thinking more efficiently than humans sparked both fascination and concern. Even though the Turk was not intelligent per se, it aligned well with the vision of a probable future in which machines could think on their own and might even end up taking over for humans.

Unfortunately—or maybe fortunately—the Turk turned out to have more in common with the steam engine than with humans, given that it was not actually capable of making its own chess *decisions*, but instead was simply a mechanical tool that could move pieces around. Both the Turk and the steam engine required humans to operate them and were not capable of making decisions on their own.

Although a long time would pass before machines making their own decisions became reality, I like to think that people in the 19th century were still intrigued by the idea. Maybe they read about "the engine" in *Gulliver's Travels* by Jonathan Swift and had strong opinions on whether a *thinking machine* could be built. And perhaps the most enthusiastic among them followed tech news closely and took note when the world saw its first *algorithm* emerge in the early 19th century.

Algorithms

Yes, algorithms existed long before artificial intelligence. Despite the fact that, in modern parlance, we use the word to describe computer programs that analyze our data or affect our behaviors, algorithms are much more than that. Put factually, an algorithm is a set of instructions carried out in a specific order to achieve a goal, with the classic example being a recipe. If a recipe does not provide all the instructions needed, you won't end up with a dish. If the order of the steps is mixed up, the recipe is also impossible to follow. As humans, we are quite good at creating algorithms for one another: Whenever you explain how something is done, you're essentially creating an algorithm. If I need to borrow some flour from you, you might answer, "Sure, you can, but I'm not home, so you'll have to let yourself in. Use the key under the mat, go into the kitchen, and look in the bottom drawer next to the fridge." In doing so, you would enable me to borrow flour from you, and I would thank you for creating the algorithm that made it possible.

Algorithms make computers work, and no computer can work without instructions from algorithms. Developing good algorithms for computers is far from easy because machines need each step to be spelled out clearly—and because the best way to solve a given problem isn't always, or even

often, obvious. At tech companies that are considered "cool" to work for, interview questions often involve describing or crafting good algorithms on the spot.

You might not believe me, but creating algorithms is among the most fun things you can do—not because explaining things to a machine is so much fun, but because thinking about and devising the best way to solve a problem is thrilling. Countess Ada Lovelace felt the same way. She quickly grew tired of typical countess pastimes and instead devoted her life to computer science. In 1843, she wrote the world's first algorithm for computers; because of this accomplishment, she is widely considered the world's first programmer. Together with the inventor Charles Babbage, she worked tirelessly on designing the world's first mechanical computer. It was never built, however, because the research funds dried up and because Babbage quarreled with his chief engineer. But although the computer itself did not materialize in their lifetimes, Lovelace is still credited with inventing the algorithm, and the idea of a machine performing a process of any kind was born.

Analog Computers

The *idea* of computers existed long before the first computers were ever built, and the first ones actually built do not resemble the computers we use today in the slightest. The most profound difference is probably that the first computers were *analog*: They were built to perform one specific task. To solve any other task, they would have to be rebuilt. When this kind of computer was built to solve a mathematical problem, it became a physical manifestation of the mathematical problem. This might sound abstract and difficult to understand, but I'm willing to bet that you've used an analog computer, possibly quite recently. Children often learn to add and subtract using counting frames, where they slide colorful beads along thin rods. We humans have used abacuses to perform calculations since antiquity, and while many might consider it a stretch to call them machines, they still illustrate how analog computers work: First, they take in analog data, namely physical quantities, directly from reality. This data doesn't need to be converted into numerical values or approximations but instead

goes directly into the machine. The analog machine then performs calculations directly by measuring the continuous changes to the physical quantities and presents the calculation result in a manner that can be directly observed. A miniature version of this is a mercury thermometer, where we do not read a number off a display but instead carry out the measurement ourselves by seeing how far up the tube the quicksilver has risen. Even more advanced analog computers that we encounter in everyday life are mechanical wristwatches and speedometers. If you should remember one thing here (because it will matter later), it's that, for analog computers, the software and the computer itself—the hardware—are one and the same.

The advantage of analog computers is that, since they receive their data directly, measurements don't need to be converted into numbers, with all the complications and potential errors that that process can contain. However, the major drawback of analog computers is that they can only solve the exact problem for which they were built.

Gödel, Turing, and von Neumann

Now we will meet three mathematicians who were indispensable in the development of modern computer science and whose contributions give both researchers and students headaches to this day: Kurt Gödel, Alan Turing, and John von Neumann. In 1931, Gödel presented a proof that shook the world so profoundly that its impact is still felt today. He demonstrated that no mathematical systems can be both *complete and consistent*, as a mathematician would put it. In other words, it's impossible to create a logical system in which everything that is true can be proven and which is also completely free of contradictions. Think about the statement "This sentence is a lie." The more you think about it, the stranger it becomes that such a paradoxical statement can even be created. This kind of thinking is exactly what Gödel did: He thought about systems that make self-referential statements and ended up showing that such statements spell trouble in the world of mathematics.

Alan Turing was deeply intrigued by Gödel's proof and wondered what this would mean for computers. In 1937, Turing proved two things, both of which are still considered landmarks within computer science. One of the

things he proved was that it's impossible to create a computer program that can determine whether or not an algorithm can be executed, that is, whether a computation will ever finish running. This is referred to as the *halting problem*, and even though it may seem like a rather odd thing to worry about, we run into it everywhere. For example, let's say you have an antivirus program and want to know whether it will ever carry out a malicious action against your computer. This isn't something you can determine; you've just encountered the halting problem and have to give up (just like Turing in 1937).

The proof that there is no proving if an algorithm will ever finish is wonderfully cunning and completely in Gödel's spirit. Let's assume that a program that can find out whether an algorithm will eventually finish *does* exist. We can call this program `WILL_STOP`. Next, we can write the following, little computer program:

```
INGAS_ALGORITHM:
    If INGAS_ALGORITHM ever finishes:
        Execute this instruction forever.
```

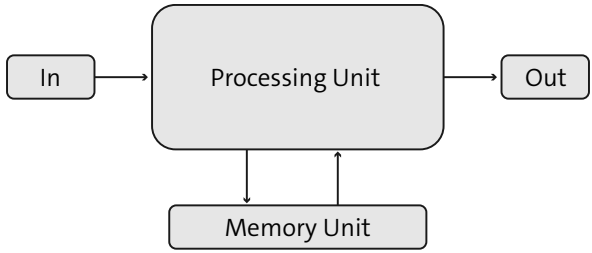
What can the program `WILL_STOP` now say about `INGAS_ALGORITHM`? If `INGAS_ALGORITHM` is going to finish, `INGAS_ALGORITHM` will have to wait forever—in other words, it will never finish. We have reached a self-contradiction (and possibly, a bit of a headache), and in doing so, also proven that the program `WILL_STOP` cannot exist.

The second thing Turing proved was that machines capable of performing any mathematical calculation can exist, as long as the calculation is formulated as a set of instructions—in other words, as an algorithm. This theoretical machine has been given the name the *Turing machine*. The Turing machine introduced the idea of a separation between the machine that performs calculations (what we know as hardware) from the computations themselves, which we refer to as software. While this idea makes perfect sense to us 21st-century humans accustomed to installing new programs on old computers (not to mention installing new apps on the same phone), the separation between hardware and software was anything but obvious

in the 1930s. The only calculation machines that existed before Turing were the analog computer and the human brain. For both of these "machines," hardware and software are the same thing. Although perhaps not obvious at first (or second) glance, Turing's two proofs form a major part of the foundation on which modern computers rest.

Even though the Turing machine was ever so brilliant, it was still just an abstract concept. Turing proved that such a machine *could* exist in theory, but as we know, the road from idea to product can be long and twisting. To build a machine that can perform any kind of calculation, concretization is required, which is where our third hero comes in: John von Neumann. He designed the structure through which we still build modern computers, so the next time you turn on your computer, you can send John a warm thought and thank him for the *von Neumann architecture*. In this architecture (or structure), computers have one central component that performs computations and another component that stores information. Consequently, when it's time to perform computations, the necessary ingredients must be collected from the memory unit, and if the result of a computation is to be stored, it must be moved from the processing unit into the memory unit. Your computer works in the same way: Most things are safely stored in the memory unit, and only what is needed to perform a computation is moved to the processing unit. Of course, in today's computers, a lot of additional stuff happens between the processing unit and the memory unit, for instance, to make storage and computations faster and easier. But if we squint a little, modern computers still look the way von Neumann drew them back in his day.

The last ingredient in the von Neumann architecture is two units that enable humans to communicate with the computer. One unit receives commands from the human and feeds them into the processing unit, while another extracts information from the processing unit, thus enabling the human (or another computer, for that matter) to receive it. The following figure is a simplified version of the von Neumann architecture and, in some sense, an illustration of your computer.



The von Neumann architecture was a fundamentally new way to design computers, and it revolutionized computer science. It described a computer capable of processing any set of instructions, as long as it could interpret them, and capable of performing calculations based on those instructions. The first von Neumann machine was built in 1952. That same year, mathematician Grace Hopper invented the first program that enabled computers to translate words—specifically English words—into machine language. Before that, computers had to be programmed directly using 0s and 1s. Writing 0s and 1s isn't enjoyable for us humans, who are used to expressing ourselves with words. With this in mind, what Grace Hopper created was the world's first *compiler*.

Even in the 1950s, before modern computers were even fully invented, researchers pondered whether providing *precise instructions to computers was the key to making them intelligent*. Of particular note is one answer John von Neumann gave during a lecture that illustrates his stance: An audience member had commented that it's impossible for a machine—at least a man-made machine—to think. To this, von Neumann replied: "You insist that there is something a machine cannot do. If you will tell me precisely what it is that a machine cannot do, then I can always make a machine which will do just that!" It is a delightfully charming fact that, while we are still building our computers according to the von Neumann architecture, we also still do not know whether they can embody actual intelligence.

With the development of computers in the mid-20th century, artificial intelligence finally moved away from the land of fraudsters and dreams, as mathematicians and scientists began to formalize concepts and ideas into theories, bringing artificial intelligence into the realm of modern science.

And so, the pursuit of creating computers that think like humans was underway.

A Field of Study Is Forged

Few academic disciplines have as clearly defined a beginning as artificial intelligence. It all took place at a small workshop at Dartmouth College, on the East Coast of the United States. The workshop was organized by the young researcher John McCarthy, who would later become one of the most unquestionably influential scholars in the field of artificial intelligence. Before starting at Dartmouth, McCarthy studied mathematics at Princeton, where he met another rising star within the field of artificial intelligence: Marvin Minsky. The two quickly hit it off and realized they shared a passion for the question of whether or not computers could possess intelligence. In 1955, McCarthy convinced Minsky and two other fellow students to help him organize a small workshop. They envisioned a two-month gathering during which ten people, in the summer of 1956, would figure out this "intelligent computers" business. In the wake of this conference, the term *artificial intelligence* was coined (in a paper by McCarthy and his three colleagues)—mainly to distinguish the field from the related field of cybernetics.

The difference between artificial intelligence and cybernetics is subtle, yet fundamental. While both fields involve studying intelligent systems, the purpose of artificial intelligence is to create machines that mimic human behaviors and human intelligence. Cybernetics, on the other hand, is about understanding how intelligent systems—whether biological or mechanical—process information. More succinctly, cybernetics is the study of communication and control, while artificial intelligence, was—and still is!—the attempt to create intelligence in an artificial system. After the name had already become established, McCarthy later disclosed that no one really loved the term *artificial intelligence*, but in the end, they needed a name, so they were stuck with it.

Together, the four students applied for research funding in the summer of 1956 to finance their workshop. In the application, they listed reasoning, machine learning, neural networks, creativity, and language comprehen-

sion among the topics they wanted to explore. They wrote 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." And so, the discipline was founded. Even though it all came together quickly at the time, the groundwork that McCarthy and his fellow students laid that summer remains the foundation of modern research in artificial intelligence. To this day, we are still studying the same concepts—machine learning, neural networks, language comprehension, and so on—and the fundamental idea that learning and intelligence can be imitated by computers continues to be the dominating viewpoint in the field.

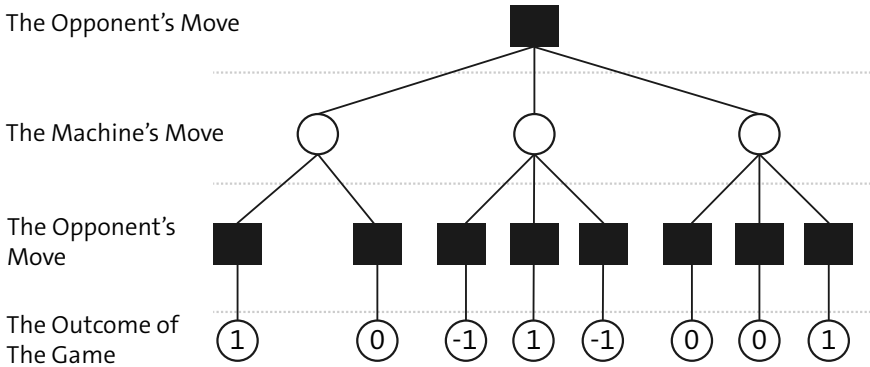
That an entire field can be founded and defined for many decades by a group of happy students over just one summer must either mean that these students are exceptionally talented or that the field is considerably more difficult than expected. The truth is probably a bit of both. The four students—John McCarthy, Marvin Minsky, Herbert Simon, and Allen Newell—became pioneers in the field and can safely be considered "the four greats" of artificial intelligence. Still, this was only the first of many times that we humans would underestimate how complicated the road to artificial intelligence would be. Time and time again, we believed ourselves on the verge of a major breakthrough that would make machines think independently, and time and time again, we have come to realize the opposite.

As the students held their summer workshop, they made plans with great optimism. Yet, 20 years later, in the 1970s, we were still miles away from solving the riddle of machine intelligence. There were, for example, still no computers capable of beating good chess players. We had programs that could follow the rules and challenge amateurs, but that was it. In fact, not until the mid-1990s did chess masters face real competition from machines.

Checkmate

While it's easy to make a machine understand the rules of chess, teaching it to select the best move from the wide array of all possible moves is incredibly difficult and computationally demanding. Why? Because mastering chess is all about knowing what will happen next and choosing your moves accordingly. To pick the next move in a game, a chess computer

must build a large search tree containing all possible future moves by both players—itself and its opponent. The following image shows an example of a tiny part of a search tree, where white circles represent the machine's moves and black squares represent the opponent's moves:



At the bottom of the tree, we find the outcomes of all possible games: 1, 0, and −1 for a win, a draw, and a loss, respectively. Given such a search tree, a chess computer can determine which move is the best using an algorithm called *Minimax*. This algorithm was created by John von Neumann in 1928—yes, long before artificial intelligence was even a discipline—and adapted in 1950 for chess by Claude Shannon (who also attended the famous Dartmouth conference). The algorithm works by moving upward from the last move and selecting the best position for each possible position that the opponent might take—that is, the maximum. On the next level, it assumes that the opponent will choose the worst possible position for itself, in other words, the minimum. This maximum/minimum stuff is the origin of the algorithm's name: Minimax. Finally, the best among the three highest-ranked positions is chosen as the best move. Next, it's the opponent's turn to make a move, and then … the chess computer must build an entirely new tree based on the new position on the board and repeat the whole Minimax process. And so it goes, move by move, until the game is over. To say the least, Minimax is a clever idea, but if you stare at a chessboard for a bit, you quickly see a huge problem: It's impossible to imagine every possible game that can unfold on the board—or search through the entire tree—as a chess computer would have had to do. If, on

average, it's possible to make 35 moves from a given position and a game of chess lasts for around 80 moves, we end up with a search tree of 35^{80} possible positions. That number is beyond huge. We can write it as 10^{123}, meaning a 1 followed by 123 zeroes. That's how many positions Minimax would have to process to say anything about the value of a position. By comparison, the entire universe, with all its interstellar dust and every known galaxy, contains roughly 10^{75} atoms. This means two things. First, it's unlikely that chess will ever be solved with complete accuracy. We cannot know whether White can win by playing perfectly or whether a perfect game would end in a draw. In that sense, chess is one of the mysteries of our universe, which is kind of beautiful. Second, since a chess computer can't possibly search through the entire tree, we have to resort to some tricks. Three tricks, to be precise.

The first trick—one that truly separates strong chess computers from weak ones—is creating an evaluation function that approximates how favorable a given position is, *without having to search the entire tree*. A moderately good chess computer might, for example, simply compare the number of pieces each player has. A slightly better chess computer will assign different weights to different pieces, recognizing that a queen is worth more than a pawn, and so on. No matter how advanced this function is, in the end, it produces a number indicating how favorable a position is.

On to the next trick: Since no computer can search through the entire tree, we need to provide it with a rule regarding how deep to search: a maximum depth. "Only look five moves ahead," the rule might say. However, this rule can cause a serious problem: Move five might involve the computer using its queen to take one of the opponent's pawns, which is good. However, what if this pawn was protected by another pawn, causing the opponent to capture the computer's queen with a pawn, in move number six? The machine would be unable to account for this event and would therefore evaluate the position five moves ahead entirely incorrectly. An algorithm called *quiescence search* was developed to solve this issue; with quiescence search, the computer will continue searching until it reaches a stable position, where serious events like "pawn captures queen" cannot happen.

And now for the final trick: Computing power is among the most valuable resources search algorithms have. To save computing power, it's important

to recognize that some future positions are outright hopeless and not worth examining. To see which future moves are worth pursuing down the tree and which branches are irrelevant, an algorithm called *alpha-beta pruning* is used. *Pruning* means to cut away or trim, and the simple purpose is to virtually trim the irrelevant branches from the tree—just like we can trim the bad branches off a fruit tree.

These ingredients—Minimax with a fixed depth, quiescence search, and alpha-beta pruning—are the cornerstones of all chess computers, including *Stockfish*, the standard chess computer for evaluating players' positions on live chess broadcasts. Today, Stockfish can beat any human chess player using only the computing power found in an iPhone. However, we haven't always had access to all this computing power.

Professor Edward Fredkin was behind many important advances in artificial intelligence. To this day, he is perhaps best known for lending his name to a research prize, announced in 1980 by Carnegie Mellon University. The Fredkin Prize, totaling $100,000, was meant to motivate computer scientists to create a computer that could beat the world's best chess player. However, the researcher who would eventually go on to win the Fredkin Prize never set out with that ambition. Instead, his victory was the result of coincidence and a PhD that went in a different direction than initially intended. The hero of this story is the young PhD student Feng-hsiung Hsu, who had little regard for artificial intelligence. Hsu began working on his PhD at Carnegie Mellon University in 1985 and later stated that, while he didn't consider artificial intelligence to be "bullshit," he had "seen some so-called research in artificial intelligence that really deserved the bullshit label." Hsu also had no particular interest in computers that played chess. He was more of a hands-on computer scientist with a fondness for engineering problems—especially those involving a computer's tiniest building blocks: chips. Simply put, a chip is a tiny flat object with an electronic circuit that, together with other components, makes up computers. And if there was one thing Hsu enjoyed, it was creating, and continuously improving, chips.

Today, you can turn your phone or computer into a chess master by buying the correct app or program, even though neither machine was built specifically for playing chess. This is the charm of digital computers: they can execute any kind of program without being built specifically to do so. But in

the late 1980s, chess computers faced a major computing capacity challenge: Even given the three tricks, a chess computer is more successful the more potential future positions it has the capacity to explore. The best chess computers were therefore specifically designed and built for fast searches. And one key ingredient in custom-built computers is high-quality chips. This is how Hsu and his fellow students were recruited—though somewhat reluctantly—to participate in chess tournaments for computers. Hsu only had seven weeks' notice for his first tournament, which meant that his development happened under pressure and without enough testing. He and his team participated with a program they named *ChipTest* to emphasize that the program had not yet been fully tested. The outcome was mediocre, but less than a year later, the team won a convincing victory with an improved version of the program. At that point, ChipTest was searching through 500,000 chess moves per second. The road from ChipTest to a truly strong chess computer that could beat grandmasters would demand a significant increase in the number of chess moves the computer could search through.

When Hsu first started working with what would become ChipTest, estimates suggested that, if the speed of the hardware could be increased a thousandfold, it might be possible to create an artificial world chess champion. This engineering challenge was enough to ignite a spark in Hsu; his motivation was rooted in finding out whether a substantial increase in speed would truly be enough to solve the chess problem. For Hsu, beating the world's greatest chess player was merely a potential bonus.

After completing his PhD, which ended up being about ChipTest and chess tournaments, Hsu started working at IBM. There, ChipTest was developed further into *Deep Thought*, which sounds like something out of the *Terminator* movies. In 1989, Deep Thought challenged the reigning world champion, Garry Kasparov, for the first time, and Kasparov easily defeated Deep Thought in both games they played.

To turn his chess computer into a champion, Hsu needed to find a better, smarter way to search through *enough* possible future moves to win. And that's exactly what he did. In collaboration with many other engineers, he spent his entire PhD, followed by a decade of development, to make it happen. The work culminated in a beast of a chess computer named *Deep Blue*, which would go on to become the first machine in the world to defeat a

reigning world champion. At this point, it would be great fun to tell you about the brilliant trick Hsu came up with—one that made it possible for computers to become chess champions. However, there was no such trick. What Hsu did over the years he worked on chess computers was simply a solid, marvelous piece of engineering, combined with sound algorithm design. He built custom-designed chips and worked intensely to make the chess algorithms as computationally efficient as physically possible. He also collaborated with grandmasters Miguel Illescas, John Fedorowicz, and Nick de Firmian to fine-tune the evaluation functions. Deep Blue contained vast amounts of chess knowledge, in addition to the advanced algorithms and computing power Hsu had spent his entire career building. It all paid off, and on February 10th, 1996, Deep Blue defeated the reigning world champion Garry Kasparov. Still, this is not the famous date: Kasparov and Deep Blue didn't meet in a match adhering to the official chess tournament rules until the spring of 1997. After six games, played over eight days, Deep Blue became the undisputed winner. May 11th, 1997, became a significant milestone in the history of artificial intelligence: For the first time, humans were dethroned in chess by a machine.

The Drosophila of Reasoning

Garry Kasparov had competed against chess computers several times before the fateful matches. In 1985, Kasparov played against 32 of the world's best chess computers at the same time—and defeated them all. As Kasparov later stated, "To me, this was the golden age. Machines were weak, and my hair was strong!" Twelve years later, Kasparov had his hands full focusing on just one chess computer—which ended up defeating him. By then, Kasparov had been world champion for over ten years. Before the match against Deep Blue, he had asked to study other games which Deep Blue had played, but IBM declined. Kasparov was used to assessing his opponent's strategy, observing their body language, and looking them in the eye. Sitting across from Deep Blue gave him a new and unsettling feeling. I imagine that factory workers who saw their output outstripped by the steam engine during the First Industrial Revolution may have had the same feeling. On the cover of *Newsweek*, the match between Kasparov and Deep

Blue was referred to as "the brain's last stand." After his defeat in 1997, Kasparov began to wonder whether Deep Blue was invincible and whether chess had now been solved, conquered—finished. Later, he disclosed that he felt doubt and fear.

Perhaps he even felt an existential fear. As Kasparov has pointed out, Deep Blue likely didn't feel the same fear. Hsu had an entirely different view on the matter. He said, "The contest was really between men in two different roles: man as a performer and man as a toolmaker." Deep Blue had neither intuition nor experience—both of which Kasparov had in impressive amounts. Deep Blue won based on computing power, advanced search algorithms, and evaluation functions developed by human experts over several decades. In short, Deep Blue calculated its way to victory. Today, no one is frightened or surprised by the idea of computers defeating chess masters, any more than we are shocked by a motorcycle moving faster than Usain Bolt. Deep Blue searched through a staggering 200 million positions per second. This process bears little resemblance to how humans play chess—just as airplanes don't flap their wings like birds. Deep Blue's method is what we in programming call *brute force*: Using raw computing power, it calculated its way to the best move.

Still, the dominance of machines in chess has not made the game any less interesting to us humans. Although today, any smartphone can run chess apps more powerful than Deep Blue, we continue to follow our human chess masters with enthusiasm—and we play more chess than ever. In 2018, Kasparov pointed out that chess, in addition to being a fascinating game, is considered the *Drosophila of reasoning*. Because just like the *Drosophila melanogaster*—also known as the fruit fly—is a useful organism for genetic researchers, chess has become what is known as "a laboratory of cognition." In fact, I think we can safely say that chess is the most extensively studied domain within artificial intelligence. In the 2020s, we still use chess to develop and explore machine intelligence.

For humans, playing chess is about abstract thinking, strategy, pattern recognition, deep focus, and often psychology as well. Humans do not master chess by evaluating millions of positions per second. There is something more, something we humans have *understood*. And that is exactly what we are looking for in intelligent machines: this mysterious *understanding*.

The Best Chess Player in the Universe

Even though Deep Blue was a fantastic feat of engineering and played chess intelligently, few of us would consider it truly *intelligent*. This feeling is an excellent example of a historic phenomenon that is so common that it has been given a name: the *AI effect*. The AI effect describes the phenomenon we observe just about every time someone builds a computer that solves a problem that we once believed required intelligence—like playing chess—upon which we change our minds and say, "Oh yeah, but that's not intelligence! That's just a calculation!" It's hard to say what it would actually take for us to be convinced of a machine's intelligence. Does it have to be adaptable, or does it need consciousness? Must the calculation it performs be incomprehensible to us, akin to how the human brain is still a mystery to us? Think about it. If you settle on the last criteria—that the machine has to do something slightly mysterious—you are not alone.

In the 1980s, chess master and philosopher Elliot Hearst analyzed just how differently humans and computers behave when playing chess. To calculate the probability of winning for each position, chess computers use smart algorithms and lightning-fast searches through vast numbers of possible outcomes for potential moves. Human chess players, on the other hand, do something entirely different, as Hearst's experiments showed. What humans do is both faster and more efficient: We recognize patterns on the chessboard. Pattern recognition is how the best chess players among us can look directly at a position and determine which strategy is the most appropriate. Humans with a deep understanding of chess have, in other words, grasped something overarching—something conceptual—that is hidden in the combinations of pieces and that has little to do with searching through millions of possible moves. Hearst argued that, unless computers learn how to recognize patterns and understand abstract concepts in the same way, they will never be able to play chess like a human.

One way of thinking is that traditional chess computers, like Deep Blue, win through sheer computing power, and their capacity to evaluate an enormous number of positions compensates for their lack of intuition about the game. But we can also reverse our thinking: Since even the most skilled humans don't have the capacity to think more than 10 to 20 moves ahead, we've been forced to develop another skill to play chess—namely,

intuition. More recently, developments in artificial intelligence have led to chess computers capable of mastering a combination of the two: both rapid search and pattern recognition that resembles an intuition for the game.

On December 5th, 2017, *AlphaZero* was introduced to the world. This is a chess computer of the most modern kind, created by one of the world's leading research environments in artificial intelligence—the company DeepMind (which was acquired by Google in 2014). It didn't take decades to develop AlphaZero; in fact, after just 24 hours of playing against itself, it had *taught itself* to play chess on a superhuman level. According to DeepMind, AlphaZero not only played at a superhuman level; it also outperformed leading chess computers like Stockfish. AlphaZero learned to play chess by training on its own. That's where the "zero" in its name came from. It knew nothing—zero—when it started playing against itself. And over the course of just 24 hours, it developed into the best chess player in the universe (unless someone has invented chess on another planet, that is). We do have to remember that 24 hours for a computer is not the same thing as 24 hours for a human. DeepMind—and all major tech companies—have access to huge amounts of processing power. Today's version of AlphaZero has played more than 44 million games, allowing it to perfect its technique. Playing so many games would take months on the kind of computers most of us can afford.

What AlphaZero learned through those 44 million games was, essentially, a kind of intuition—whereas Deep Blue relied on human-designed algorithms and raw computing power to win its matches. AlphaZero doesn't pick the next move directly; instead, it decides—it intuits—which moves are worth examining more closely. Based on that selection, the next move is then determined using the same kind of tree search that Deep Blue and Stockfish use. In other words, AlphaZero combines *artificial intuition* with processing power. It's estimated that the best human chess players can perform, at most, hundreds of searches; that is, they can imagine a maximum of a few hundred possible future games for each position. Today, Stockfish and other traditional chess computers perform millions of searches per position, while AlphaZero, thanks to its chess intuition, "only" needs to perform tens of thousands of searches.

Interestingly, AlphaZero has developed a non-human playing style. While traditional chess computers are based on expert (human) knowledge, AlphaZero has learned to play chess on its own, without regard to any

strategies humans have developed over the years, whether well known or esoteric. During training, it played whatever move it considered the most promising. In this way, AlphaZero discovered strategies we humans hadn't discovered—strategies that human chess players can now learn by playing against AlphaZero. In my opinion, this is the best kind of achievement when it comes to artificial intelligence: machines that can teach us something new and valuable. In this sense, chess has become more than a laboratory for intelligence: The game has given machines the opportunity to demonstrate that, through trial and error, they can acquire knowledge we humans do not have, and thus cannot give them.

The Age of Machines

Do we now feel, in a world where AlphaZero has both developed an intuition for chess and also discovered entirely new chess strategies, that true artificial intelligence has finally materialized? Most people I know would answer *no* to that question. Why so, when our artificial chess masters are so good that even Norwegian chess grandmaster Magnus Carlsen learns from them? Maybe chess boxing can help us investigate this feeling (because yes, chess boxing is a real thing). The sport involves—as you might imagine—both boxing and chess, and you can win either by physically beating your opponent or by intellectually beating them at chess. We know that a computer running AlphaZero has neither a body nor the ability to stand up; it cannot put on boxing gloves or step into a ring. AlphaZero is "just" a computer program, with a narrow set of abilities. It can play chess, but it knows nothing about the world—at least not in the way we humans understand that we live in a big world that contains so much more than chess. So far, no computer has been given the opportunity to face a human in a chess boxing tournament, likely more due to ethical reasons than practical ones. Machines that can knock humans out already exist. We could easily have taken one of Boston Dynamics' humanoid robots—the ones that impress us in YouTube videos by doing backflips and bouncing through obstacle courses—uploaded a version of AlphaZero to its computer, and used it for both playing chess and boxing. But what would that leave us with? Sure, we

would have a chess-playing boxing robot. Would it be intelligent and adaptable in the same way a human is? No, something would still be missing. But exactly what it would take—in terms of technical abilities and how those abilities are combined—for us to agree that a machine is approaching the human ability to generalize? It's hard to say.

Still, this does not mean that machines are inferior to us humans—because we're no longer the ones sitting on the intellectual throne. By the end of 2018, computers had beaten us at most games: chess, Texas Hold'em poker, Go, and even less commonly known games like Shogi and Dota 2. The latter is a very popular (and quite complex) multiplayer online game played with teams consisting of five players who must strategize and coordinate in order to outsmart their opponents. On June 6th, 2018, several artificially intelligent players managed, for the first time, to cooperate and defeat a team of human players. The reassuring news is that these artificially intelligent players are willing to work alongside human players—that is, to be partners rather than opponents. The reason why machines can beat us at our own games is not that they have an inherent desire to outcompete humans: Machines defeat us because we ask them to. It's not the machines, but their creators—researchers like myself—who work hard to build machines that can surpass humans in any domain we can think of. The fear we might have about the actions of machines is, at its core, based one of two things: either a fear of human intentions or a fear about our limited understanding of what we are asking machines to do. As a researcher in artificial intelligence, I'm not worried that my computer might have an ulterior motive for beating me at a game. I'm worried about something entirely different.

We are living in the greatest Golden Age of artificial intelligence (so far). In recent years, artificial intelligence has made remarkable strides in solving tasks, showing almost unstoppable progress. In 2017, AlphaZero was the most groundbreaking development in artificial intelligence, but in a few short years, this has changed. Artificial intelligence has really made its way out of the lab. And you don't need to be an artificial intelligence researcher to notice this progress: When you unlock your phone using your face, it's because your phone has learned what you look like—even if you change your hairstyle, get less (or more) sleep than usual, or change the lighting.

When you can't stop scrolling on social media, it's because the platform's algorithm has learned how to adapt the feed to your preferences to hold your attention for longer. When you read a text, you can't know whether it was written by a human or a machine. And when one of Amazon's warehouse workers gets fired, a machine may have made the decision. Just a few years ago, even the best chatbots struggled to carry on an intelligent conversation. Today, artificially intelligent language systems are used to create propaganda, write articles, and develop video games—and chatbots are capable of engaging humans in conversations that last for hours. In the fall of 2022, an artist smuggled a painting created by an artificially intelligent program into an art fair, and the machine *won*, without any of the judges suspecting a thing. Artificially intelligent systems have gone from clumsy to superhuman in just a few years.

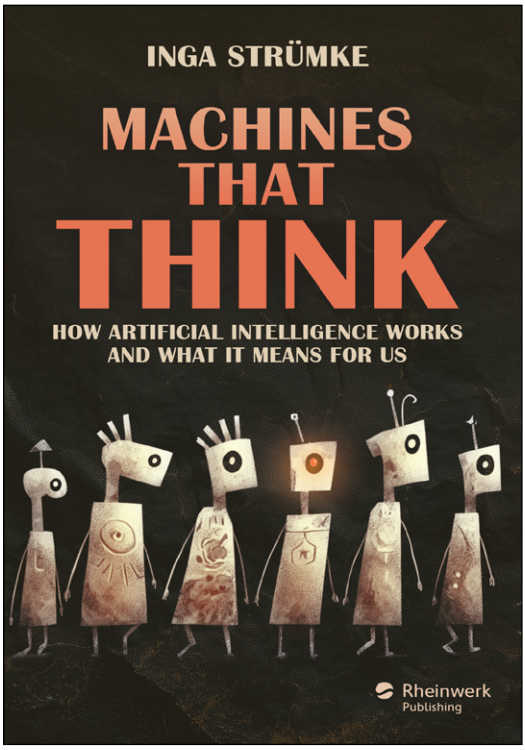
At the same time, many researchers—including myself—believe that artificial intelligence is nowhere near achieving either consciousness or general intelligence. That even the best chatbots are mechanical parrots that recycle whatever we have fed them. And that the real challenge is not that we will be sharing the planet with another intelligent species, but the fact that we are increasingly surrounded by systems that reinforce our own biases and prejudices. Regardless of how long it may be before we achieve artificial general intelligence, artificially intelligent technology is already making its way into society, into our lives—and into our pockets. In a few years, most of the images you see online may be created by machines, and many of the conversations you have online may involve only one human: *you*. Moreover, tech companies may influence political opinions by boosting specific arguments on social platforms, as well as by employing well-paid lobbyists in political decision-making arenas. Because of this, I believe that basic knowledge about artificial intelligence will quickly become crucial for understanding much of what is happening in our lives and for being able to actively participate in the public debate about how we want this technology to influence our society.

Let's begin our journey by listening to Voltaire, who said, "Define your terms, or we shall never understand one another." The very first thing we need to agree on is an abbreviation. Artificial intelligence is commonly abbreviated as AI, and for the rest of the book, I will use this abbreviation.

So, what is artificial intelligence—or AI? In the U.S., the National Artificial Intelligence Act of 2020 uses the following definition: "The term 'artificial intelligence' means a machine-based system that can, for a given set of human-defined objectives, make predictions, recommendations or decisions influencing real or virtual environments."[1] At the risk of being quarrelsome, that definition can encompass everything from humans ourselves to websites where you can look up a product—like Walmart.com. But this ambiguity (for once) is not the government's fault.

This stems from the challenge of clearly defining what "intelligence" actually is. But instead of falling down a philosophical rabbit hole, I'm going to assume that you have some idea—an intuition, some might say—of what you consider *intelligence* to be, which we will lean on. Next, we can say that *artificial intelligence is a field within computer science with the goal of developing machines that are capable of behaving intelligently*. There you have it—the dream and the goal: to create intelligent machines. But how do we go about making machines intelligent? Fundamentally, there are two approaches that are so different that they form a deep philosophical divide within the academic discipline: the *symbolic* and the *subsymbolic* approaches.

1. *https://www.congress.gov/bill/116th-congress/house-bill/6216*



Inga Strümke

Machines That Think

How Artificial Intelligence Works and What It Means for Us

- The journey to AI, from the first algorithms to modern applications
- Ethical and social implications of AI
- The future of machine intelligence

rheinwerk-computing.com/6171

Inga Strümke is a Norwegian physicist specializing in artificial intelligence and machine learning. She was born in 1989 in Gummersbach, Germany, and grew up in Narvik, Norway. Strümke holds a master's degree in theoretical physics from Norwegian University of Science and Technology (NTNU) and a doctorate in particle physics from the University of Bergen. She is currently an associate professor at NTNU.

Strümke is also known for her work in AI ethics and has received an award for science communication from the Norwegian Research Council.

ISBN 978-1-4932-2761-7 • 256 pages • 01/2026

E-book: $24.99 • **Print book:** $29.95 • **Bundle:** $39.99

Rheinwerk Publishing