The Book Review Column [1]

by **Nicholas Tran** (ntran@scu.edu)
Department of Mathematics & Computer Science, Santa Clara University

1 Notable New Releases

Mathematical Biology (The MIT Press, 2025), by Markus Meister (Caltech), Kyu Hyun Lee (UCSF), and Ruben Portugues (Technical University of Munich), covers the principles and applications of linear algebra, probability and statistics, and dynamical systems. The intended audience is advanced undergraduates and graduate students in the biological sciences.

A Geometrical Introduction to Tensor Calculus (Princeton University Press, 2025), by Jeroen Tromp (Princeton University), provides a geometrical understanding of tensors and tensor calculus, which are essential for understanding deep learning algorithms, from the point of view of a physicist.

What is Mathematical Logic?, 2nd ed. (Oxford University Press, 2026), by Guillermo Badia (University of Queensland), John N. Crossley (Monash University), and John C. Stillwell (University of San Francisco), is a concise and accessible introduction to mathematical logic, updated to include new material on automatic theorem proving and intuitionistic, free, and modal logics.

Linear Cryptanalysis (Cambridge University Press, 2025), by Tim Beyne and Vincent Rijmen (both of Katholieke Universiteit Leuven), provides a comprehensive and in-depth treatment of linear cryptanalysis, requiring no prior knowledge of cryptography. The most influential papers on the subject are presented in a consistent framework based on linear algebra.

Guide to Using Generative AI in Programming (Springer, 2026), by Antti Laaksonen (University of Helsinki), explores how generative AI tools can be applied effectively to programming tasks and examines how learning and teaching programming are evolving in the era of ChatGPT.

2 This Column

Lance Fortnow reviews the controversial book *If Anyone Builds It, Everyone Dies: Why Superhuman AI Would Kill Us All* (Little, Brown and Company, 2025), by Eliezer Yudkowsky and Nate Soares. He challenges theorists to develop a mathematical framework for evaluating the capabilities of AI.

Bill Gasarch recommends the accessible and practical book *Math for Security: From Graphs and Geometry to Spatial Analysis* (No Starch Press, 2023), by Daniel Reilly, which demonstrates how real-world security problems can be solved using existing mathematical tools while also stimulating new theoretical questions.

[1]©2025 Nicholas Tran

The Language of Mathematics: The Stories behind the Symbols (Princeton University Press, 2025), by Raúl Rojas (translated by Eduardo Aparicio) is a scholarly yet entertaining exploration of the origins of mathematical symbols. I think it would make excellent supplementary reading for an introductory course in calculus or discrete mathematics.

3 How to Contribute

Make a New Year's resolution to write a book review for *SIGACT News*. Either choose from the books listed below or propose your own. In either case, the publisher will send you a free copy of the book. Guidelines and a LaTeX template can be found at `https://algoplexity.com/~ntran`.

BOOKS THAT NEED REVIEWERS FOR THE SIGACT NEWS COLUMN

Algorithms & Complexity

1. Vaze, R. (2023). *Online Algorithms.* Cambridge University Press.
2. Bosc, P., Guyomard, M., & Miclet, L. (2023). *Algorithm Design: A Methodological Approach - 150 Problems and Detailed Solutions.* Routledge.
3. Brody, J. (2025). *The Joy of Quantum Computing: A Concise Introduction.* Princeton University Press.
4. Dalzell, A., & McArdle, S., & Berta, M., & Bienias, P., & Chen, C.-F., & Gilyén, A., & Hann, C., & Kastoryano, M., & Khabiboulline, E., & Kubica, A., & Salton, G., & Wang, S., & Brandão, F. (2025). *Quantum Algorithms: A Survey of Applications and End-to-end Complexities.* Cambridge University Press.
5. Erciyes, K. (2025). *Guide to Distributed Algorithms: Design, Analysis and Implementation Using Python.* Springer.
6. Morazán, M. T. (2025). *Programming-based Formal Languages and Automata Theory: Design, Implement, Validate, and Prove.* Springer.

Computability & Logic

1. Downey, R. (2024). *Computability and Complexity: Foundations and Tools for Pursuing Scientific Applications.* Springer.
2. Badia, G., Crossley, J. N., & Stillwell, J. C. (2026). *What is Mathematical Logic?, 2nd ed.* Oxford University Press.

Programming

1. Lichtman, E. (2025). *The Computer Always Wins: How Algorithms Beat Us at Our Own Games.* The MIT Press.

2. Laaksonen, A. (2026). *Guide to Using Generative AI in Programming.* Springer.

Miscellaneous Computer Science & Mathematics

1. Chayka, K. (2024). *Filterworld: How Algorithms Flattened Culture.* Doubleday.
2. Valiant, L. (2024). *The Importance of Being Educable: A New Theory of Human Uniqueness.* Princeton University Press.
3. Wilson, R. (2025). *Sum Stories: Equations and Their Origins.* Oxford University Press.
4. O'Rourke, J. (2025). *The Mathematics of Origami.* Cambridge University Press.
5. Meister, M., Lee, K. H., & Portugues, R. (2025). *Mathematical Biology.* The MIT Press.

Data Science

1. Alpaydın, E. (2025). *Fundamentals of Probability and Statistics for Machine Learning.* The MIT Press.
2. Tromp, J. (2025). *A Geometrical Introduction to Tensor Calculus.* Princeton University Press.

Discrete Mathematics and Computing

1. Devadoss, S., & O'Rourke, J. (2025). *Discrete and Computational Geometry, 2nd ed.* Princeton University Press.

Cryptography and Security

1. Garfinkel, S. (2025). *Differential Privacy.* The MIT Press.
2. Beyne, T., & Rijmen, V. (2025). *Linear Cryptanalysis.* Cambridge University Press.

Review of [1]

If Anyone Builds It, Everyone Dies

Why Superhuman AI Would Kill Us All

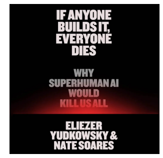
Eliezer Yudkowsky and **Nate Soares**

Little, Brown and Company, 2025

272 pages, $30 Hardcover, $14.99 Ebook, $24.99 Audiobook

Review by

Lance Fortnow



1 Summary

The lack of subtlety of the title reflects the book itself. Authors Eliezer Yudkowsky and Nate Soares lay out what they claim is an ironclad proof that if we develop Artificial Super Intelligence (ASI), then all humans will be wiped out. They then argue that we should immediately halt AI development, since we can't predict exactly when we might reach that deadly ASI threshold.

The authors are leaders of the Machine Intelligence Research Institute, co-founded by Yudkowsky in 2000 as the Singularity Institute for Artificial Intelligence. They define superintelligence as "a mind much more capable than any human at almost every sort of steering and prediction problem."

The argument builds in chapters, and so let me express what the authors do in each.

Chapter 1: The authors argue that AI has gotten far closer to human reasoning faster than any of us had expected. Agreed.

Chapter 2: Modern artificial intelligence is "grown, not crafted." It is trained on data and not by logic, and we don't understand the programs that get generated. Agreed.

Chapter 3: While the computers themselves don't have desires, they can simulate having desires, and we can reason about them as though they do. I struggle with this one and wrote a blog post[2] about it, which generated a lengthy discussion.

Chapter 4: AI will go beyond the desires we train it for and instead develop its own.

Chapter 5: AI's desires will not be compatible with human existence.

Chapter 6: A superintelligent AI is capable of killing us all.

That's Part I of the book. Part 2 describes an extinction scenario in which AI hides a killer virus in a new drug. Personally I would have gone with encouraging wars and revolutions and watching humanity kill itself off, but killer viruses also work.

Part 3 says we need to shut AI development down immediately and gives us directions for encouraging people to do so.

The book provides mostly high-level arguments, often through stories. The authors give far more detailed reasoning on their website[3], as they try to address every possible objection. On

[1] ©2025 Lance Fortnow

[2] `https://blog.computationalcomplexity.org/2025/10/computers-dont-want.html`

[3] `https://ifanyonebuildsit.com/resources`

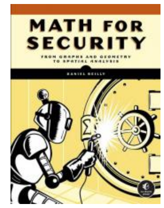
the other hand, despite this book being a bestseller and gathering many positive views, there hasn't been any significant push to slow AI development. If there are concerns about AI, they are mostly about how it affects our jobs, our intellectual property, and our education rather than as an existential threat. I would have liked to see the book balance the good that ASI could bring us with the risks, but if the authors truly believe the book's title, the good wouldn't do us any good if we weren't around to take advantage of it.

2 A Call to Arms

You should read this book for the reasons I did. As artificial intelligence's role in society continues to expand, it is essential for computer scientists, especially theorists, to understand its risks and consider various points of view, whether or not we agree with them.

While we have few tools to theoretically analyze the complexity of machine learning systems, we have developed a mathematical foundation of computing. We have an obligation to apply this knowledge to understand what computers can and cannot accomplish to help us understand the future of artificial intelligence and help shape it in the right direction. Our knowledge of computing gives us a unique frame to understand and interpret arguments about the future of artificial intelligence, such as the ones given in this book.

Review of [1]



Math for Security

From Graphs and Geometry to Spatial Analysis

Daniel Reilly

No Starch Press, 2023
292 pages, $49.99 Paperback, $39.99 Ebook

Review by
William Gasarch (gasarch@umd.edu)

1 Introduction

Seeing the title *Math for Security*, I first thought this book was about cryptography. Wrong! This book does not have any cryptography in it. Seeing the subtitle *From Graphs and Geometry to Spatial Analysis*, I first thought this book was about modeling networks as graphs. Right! Modeling networks as graphs is a major part of this book. My second thought was *Geometry?* Neither wrong nor right, since this is a question, not a guess on contents. We will discuss applications of geometry to security later in this review.

This is a very practical book for security. The math is introduced and used as needed. Reading it broke me out of my notion that the main use of math in security is cryptography.

The book is in three parts. Part I has 2 chapters, Part II has 8 chapters, and Part III has 3 chapters.

2 Summary of Contents

2.1 Part I: Environment and Conventions

Chapter 1 is aptly named *Setting up the Environment* and tells you how to install Python and some auxiliary packages onto your laptop. Chapter 2 is aptly named *Programming and Math Conventions* and goes over standard notation. Chapter 2 is indicative of the entire book in that they *do not* separate the math from the programming. They go hand in hand. This is most welcome.

2.2 Part II: Graph Theory and Computational Geometry

Chapter 3 is titled *Securing Networks with Graph Theory*. First it defines what a graph is and gives some examples. It then focuses on graphs where the nodes are either people or databases or systems.

The key question is *which node is the most important*, as that may be a node you want to attack (if you are a black hat) or protect (if you are a white hat). The chapter defines various notions

[1]©2025 William Gasarch

of *importance* rigorously. The definition of *Betweenness Centrality* really intrigued me, since I had seen it before in a paper on 3SUM-hardness and did not know it had *real* applications.

Other questions the chapter considers are *which people are talking to each other*, which is essentially finding cliques, and connectivity of a network, which is just what you think it is.

Chapter 3 is mostly theoretical. Chapters 4 and 5, titled *Building a Network Traffic Analysis Tool* and *Identifying Threats with Social Network Analysis*, use the notions in Chapter 3 to actually build tools for security: programs that analyze traffic to find which nodes (processors or similar) are most heavily used, and identifying security threats. These chapters are *not* theoretical. They give actual code (in Python), and at the end the reader can actually build these tools.

Chapters 4 and 5 use historical data to analyze a network: they can be used to find a security threat *after its happened*. Chapter 6, titled *Analyzing Social Networks to Prevent Security Incidents*, is about predictive analysis— building tools to predict security threats *before they happen*. And again, this is not theoretical— at the end the reader can actually build these tools.

Chapters 3, 4, 5, 6 use graph theory for network analysis. Chapter 7 introduces geometry as a tool for security. Geometry? Chapter 7, titled *Using Geometry to Improve Security Packages*, gives the basics of computational geometry and some applications to planning a concert and to placing guards. Placing guards is security of the old-fashioned kind: security personnel who guard doors and such. And again, they give actual code (in Python), and at the end the reader can actually build these tools.

Chapters 8, 9, and 10, titled *Tracking People in Physical Space with Digital Information*, *Computational Geometry for Safety Resource Distribution*, and *Computational Geometry for Facial Recognition*, are about applying computational geometry to modern security concerns: tracking people (the ethics of this practice is discussed briefly), allocation of security resources, and facial recognition. Chapter 9 introduces Voronoi diagrams. This is typical of the book: math is developed on an as-needed basis. And again, they give actual code (in Python), and at the end the reader can actually build these tools.

2.3 Part III: The Art Gallery Problem

The basic Art Gallery problem is as follows:

Given an art gallery, represented by an n-sided polygon, what is the minimum number of guards needed to be placed in the polygon so every point in the polygon is visible to some guard.

In this form, this seems like a nice theoretical problem. And indeed, there has been a lot of work on it by computational geometers and other theorists. As usual, this book takes some of those algorithms, makes them practical, and again, they give actual code (in Python), and at the end the reader can actually build these tools.

But that's not all. This book takes the applications to security *very seriously*. In fact, the author came across the problem because it really was the security problem he was working on. Here is the quote from the book (p. 210):

A good plan for the placement of security personnel, checkpoints, and monitoring devices can reduce the number of incidents the security team will need to respond to from the start. It can also reduce the response time when an incident does occur, thus reducing the overall risk. Unfortunately, there are often differing levels of understanding among human planners on a security team, which can lead to poorly planned (or poorly implemented) security controls. That's why I am always searching for ways to automate portions of my team's planning.

It was during one of these searches that I discovered the Art Gallery problem, which addresses the very problem I was researching: the efficient deployment of security resources for buildings with what we'll call "untraditional layouts".

The formulation of the problem I gave above is not sufficient for real world usage. Consider the following issues that the formulation ignored:

1. In the real world, art galleries and other sensitive areas are a lot more complicated than n-sided polygons.
2. The guards have a limited range of angles their eyes can see.
3. The guards cannot honestly sing the song by *The Who* which goes: *I can see for miles and miles and miles and miles and miles.*
4. Some parts of the art gallery are more important than others to cover. (Be careful here. Some modern art looks like garbage, and some is literally made out of garbage but is valuable.)
5. Getting a fast algorithm is important.

Chapter 11, titled *Distributing Security Resources to Guard a Space*, defines the problem and begins to tackle some of the issues above. Chapters 12 and 13, titled *The Minimal Viable Product Approach to Security Software Development* and *Delivering Python Applications*, are about how to make the code into a package that people can actually use.

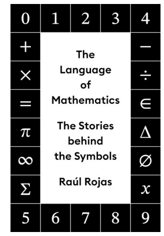
3 Opinion

This book was, for me, a real eye-opener. I am a theorist who is often skeptical about the problems theorists work on having any application. This book presents problems that have applications since:

1. The book *starts* with security needs and then sees what theory out there can help.
2. The book goes the extra n steps from a theoretical algorithm to an implementation.

Who should read this book? Both people in security and people in theory, as this book shows how theory can apply to security, and security can give new problems for theorists. The book can be read by an undergraduate who knows some discrete math and some basic programming.

Review of [1]



The Language of Mathematics

The Stories behind the Symbols

Raúl Rojas (translated by **Eduardo Aparicio**)

Princeton University Press, 2025
280 pages, $27.95 Hardcover, $19.57 Ebook

Review by
Nicholas Tran

This book traces the origins of more than 60 familiar symbols in the mathematical alphabet, providing interesting details about the lives of the people who introduced and popularized them in the process. Written in Spanish by the prize-winning computer scientist and historian Raúl Rojas and translated into English by Eduardo Aparicio, the book is an easy and enjoyable read for a general audience, requiring no technical background. Mathematicians will appreciate reproductions of original manuscripts where the symbols first appeared, whereas nonspecialists may find inspiring the author's observations about these symbols and their inventors (e.g., "[the universal quantifier ∀] resembles a cubist tear flowing from an eye that Picasso could have painted.") There are nine chapters with fifty-four self-contained sections, loosely organized by topic and in chronological order.

1 Summary

Chapter 1 traces the birth of algebra and the need for symbols back to Diophantus and al-Khwārizmī. It presents frequency counts of the top twenty letters and mathematical symbols that appear in *arXiv* mathematical texts and some engineering textbooks as evidence of their importance in modern mathematical writing. To further demonstrate the power of symbols to express deep ideas concisely, it presents the three most beautiful formulae in mathematics according to a recent survey: Pythagorean theorem ($a^2 + b^2 = c^2$), Euler's identity ($e^{i\pi} + 1 = 0$), and Euler's formula for polyhedra ($V - E + F = 2$). The chapter ends with a discussion of why we should refer to *sides* of an equation instead of its *roots*.

Chapter 2 is about number systems and variables. The evolution of our modern positional decimal system, from the Babylonian base-60 system to the Hindu base-10 system with zero added, and later to the Arabic system with the decimal point, is described, along with the base-20 Mayan system used throughout Mesoamerica. The chapter also discusses the use of letters as variables, starting with Fibonacci pioneering the use of letters to denote numbers. François Viète adopted the use of Latin consonants for constants and Latin vowels for variables. His countryman René Descartes later changed to using letters at the beginning of the Latin alphabet for constants and

[1] ©2025 Nicholas Tran

letters at the end for variables, a convention that has prevailed to this day. The chapter also discusses adorning variables with superscripts and subscripts, the absolute value bar, and the life of its inventor Karl Weierstrass.

Chapter 3 points out the use of two different types of cross symbols for arithmetic operators (and three in the British flag). It also describes the formal theory of negative integers as pairs of natural numbers and explains the German saying "nach Adam Riese" ("according to Adam Ries") referring to the father of modern calculating.

Chapter 4 presents short biographies of Robert Recorde, who invented the equal sign (=), and Thomas Harriot, who introduced the greater-than and less-than signs ($>$ and $<$). The chapter also recounts the triumph of the parentheses over the vinculum ($\overline{}$) for grouping symbols. It ends with a discussion of the history of the comma and the period as separators.

Chapter 5 delves into symbols in calculus, including the integral sign $\int$ and the differential symbol d, both introduced by Gottfried Leibniz; the partial derivative symbol ∂ created by Adrien-Marie Legendre; the nabla symbol ∇ invented by William Hamilton; the infinity symbol ∞ proposed by John Wallis; the function notation $f(x)$ popularized by Leonhard Euler; and the limit notation $\lim_{x \to a}$ introduced by John Leathem. The chapter also discusses the formal models of real numbers developed by Richard Dedekind and Georg Cantor.

Chapter 6 focuses on symbols in set theory and logic, such as the element-of symbol $\in$ and existential quantifier $\exists$ introduced by Giuseppe Peano; the universal quantifier $\forall$ invented by Gerhard Gentzen; the empty set symbol $\emptyset$, the integer set symbol $\mathbb{Z}$ and the rational set symbol $\mathbb{Q}$ created by the Bourbaki group. The chapter gives a brief explanation of Ernst Zermelo-Abraham Fraenkel's axiomatization of set theory and Georg Cantor's diagonal argument that gave rise to the hierarchy of infinities $\aleph_0 < \aleph_1 < \aleph_2 < \cdots$.

Chapter 7 examines symbols for constants such as e (Euler's number or Napier's constant), π (Archimedes's constant or Ludolphian number), i ($\sqrt{-1}$), h (Planck's constant), c (speed of light), G (gravitational constant), ε_0 (Coulomb constant) and k_b (Boltzmann constant).

Chapter 8 surveys symbols in discrete mathematics and combinatorics, including the factorial symbol ! introduced by Christian Kramp, the summation symbol $\sum$ popularized by Leonhard Euler, the binomial coefficient symbol $\binom{n}{k}$ proposed by Andreas von Ettingshausen, and the floor and ceiling symbols $\lfloor x \rfloor$ and $\lceil x \rceil$ created by Kenneth Iverson.

Chapter 9 concludes with a discussion of miscellaneous symbols such as the QED symbol $\blacksquare$ invented by Paul Halmos, the congruence symbol $\equiv$ introduced by Carl Friedrich Gauss, and the matrix notation using square brackets popularized by Arthur Cayley. It also explains the surprising origins of the terms sin, cos, and tan.

2 Opinion

The book manages to be both scholarly and entertaining at the same time, a rare feat for mathematical writing. As noted in the foreword, certain details appear several times to ensure each section is self-contained, but this repetition does not diminish the overall reading experience. I think the book would be excellent supplementary reading for an introductory course in calculus or discrete mathematics, as it provides historical context, humanizes the subject matter, and offers lucid, informal explanations of some mathematical concepts often found difficult by undergraduates such as limit, Dedekind cut, δ-ϵ definitions, diagonal argument, and axiomatic set theory.