# Theory of Automata and Languages

## Introduction

Fall 2024

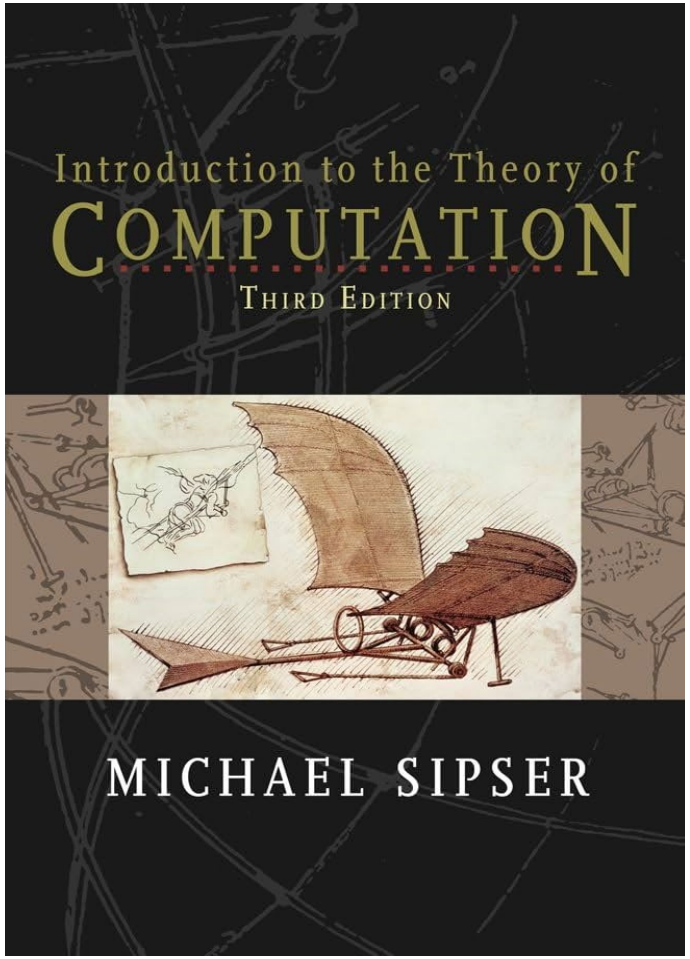
Sharif University of Technology

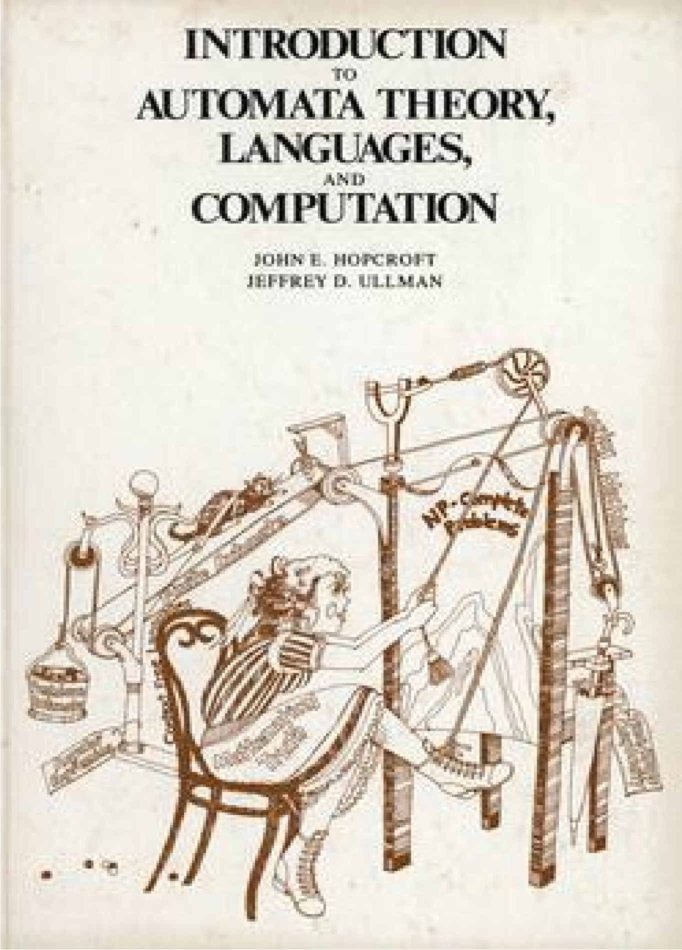
Mehran Moeini Jam

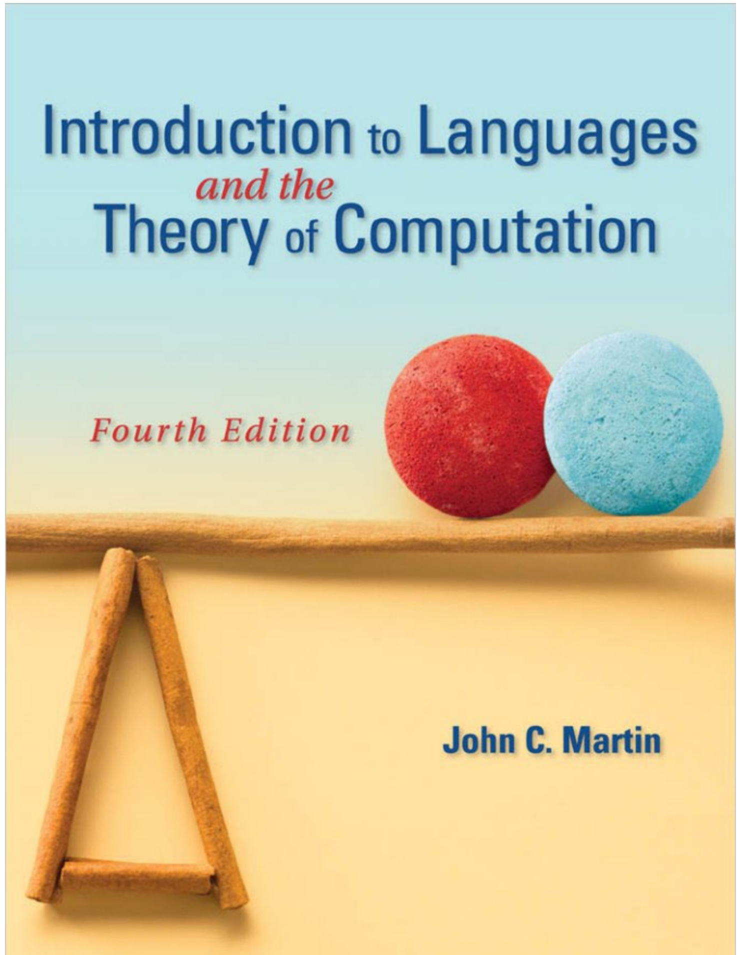
# Overview

- The School of Formalism in Mathematical Philosophy
- Learning about Theory and Models of Computation
- Problem Classification and the Boundaries of Computation
- Viewing Through the Lenses of History
- Research: The Next Steps!

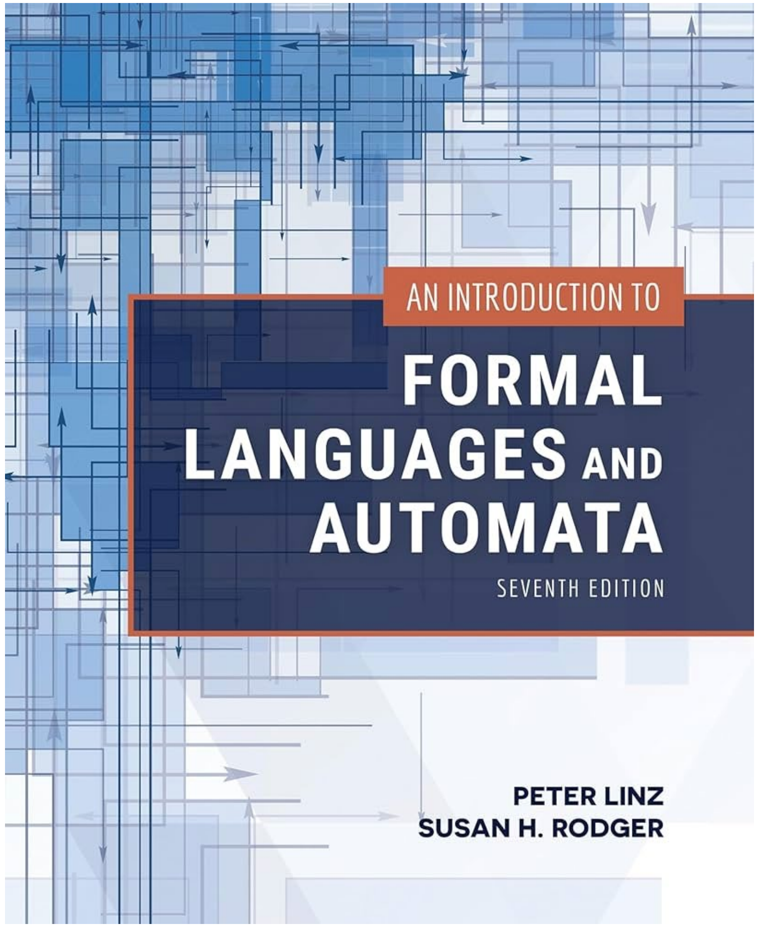
# Textbooks

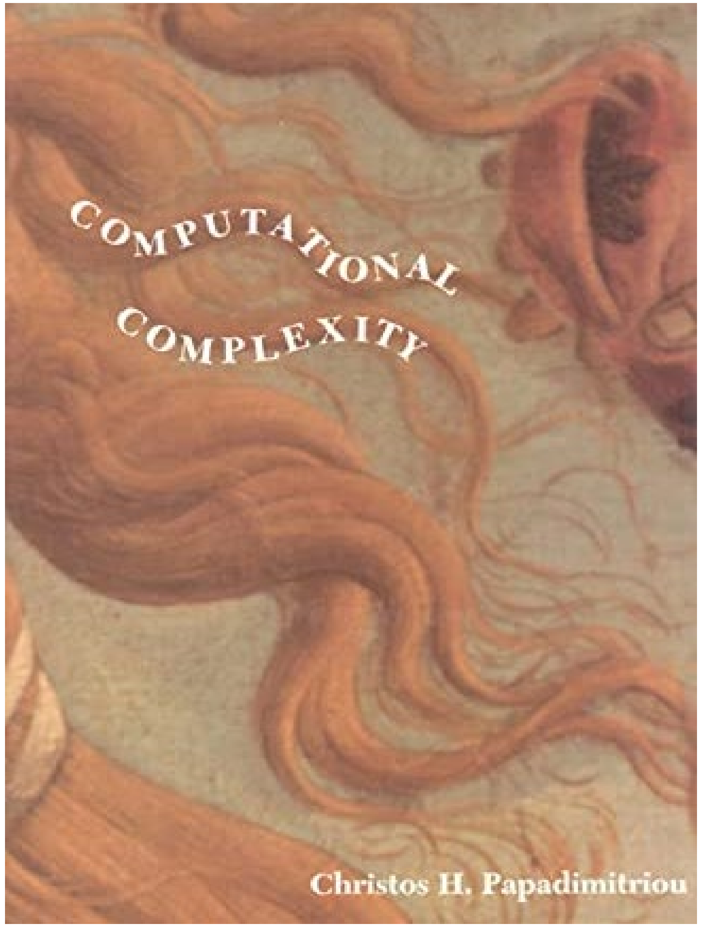

Introduction to the Theory of
COMPUTATION
THIRD EDITION
MICHAEL SIPSER


INTRODUCTION TO AUTOMATA THEORY, LANGUAGES, AND COMPUTATION
JOHN E. HOPCROFT
JEFFREY D. ULLMAN
NP-Complete Problems


Introduction to Languages and the Theory of Computation
Fourth Edition
John C. Martin

# Textbooks


AN INTRODUCTION TO
FORMAL LANGUAGES AND AUTOMATA
SEVENTH EDITION
PETER LINZ
SUSAN H. RODGER


COMPUTATIONAL
COMPLEXITY
Christos H. Papadimitriou

# The School of Formalism in Mathematical Philosophy

# School of Formalism

- I do not answer the question ‘What is a number?’ by defining number conceptually … I define from the standpoint of the pure formalist, and call *certain tangible signs* ‘numbers’. Thus the existence of these numbers is not in question.[1]

Eduard Heine, German Mathematician

# School of Formalism

- Formalists School, particularly led by David Hilbert, states that mathematics is essentially the manipulation of symbols according to some specific rules. In this perspective, proofs are sequences of symbol manipulations that follow the rules of the formal system.

- It focuses on the consistency, completeness and rigorous proofs within a formal systems, though the completeness property has been severely challenged, specially by Kurt Gödel and his famous Incompleteness Theorems.

# School of Formalism

- Formal Methods in computer science apply this formlist philosophy by using mathematical models to specify, design and verify software and hardware systems.

- The goal is to ensure that these systems behave correctly according to their specification thorough formal proofs, ensuring that no errors, ambiguities, and inconsistencies exist.

# School of Formalism

- The study of automata and formal languages plays a key role in formal methods. These concepts are rooted in formalist traditions, and are widely used in specified purposes.

- We emphasize abstraction and the mechanization of reasoning, processes that can be carried out without relying on human intuition. This principle lays the groundwork for the development of computability theory and algorithmic methods.

# Theory and Models of Computation

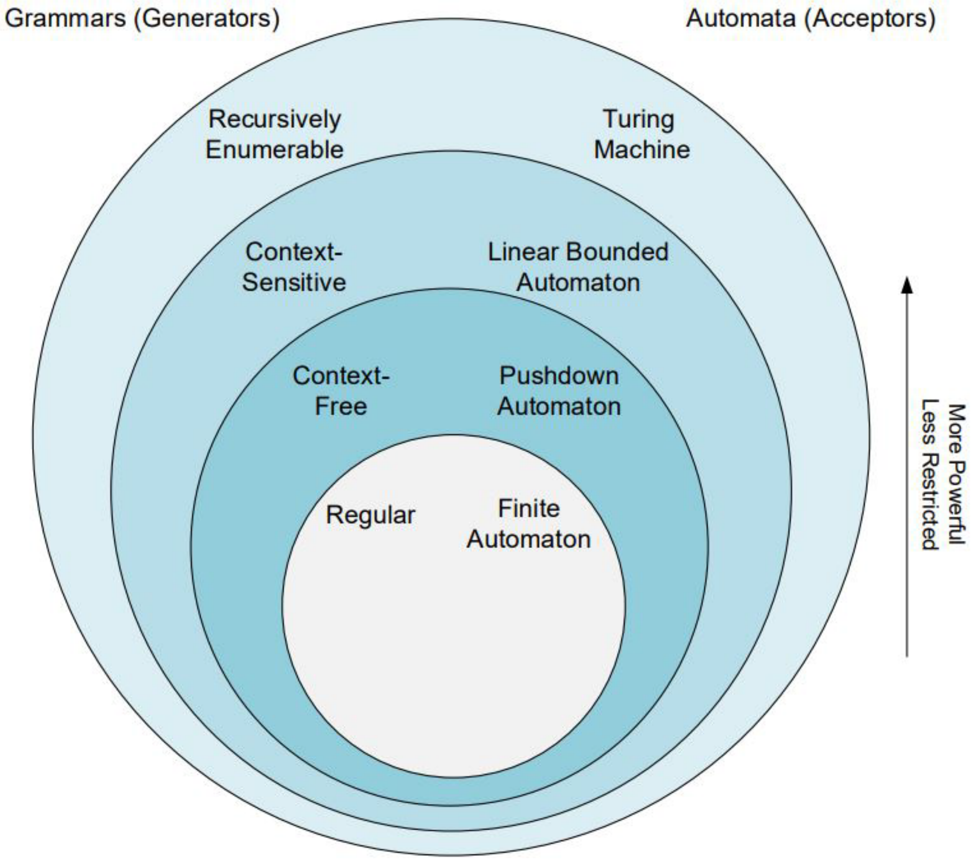
# School of Formalism

- We will explore the foundational concepts of computation and how we can model, analyze, and later verify computational processes.

- We will go through several types of models and analyze each model's strengths and weaknesses.

- This critical evaluation will help us understand their applicability and limitations in solving different computational problems.

# The Chomsky Hierarchy



# Problem Classification and the Boundaries of Computation

# Problem Classification

- In this course, we will learn that understanding *problems* in essential before pursuing *solutions and algorithms* because it allows us to accurately define the scope and requirements of the issue in hand.

- This foundational knowledge is crucial as it highlights the boundaries of computation and guide researchers in identifying areas where alternative or new approaches (such as quantum computation) may be needed.

# The Story of Hilbert’s Tenth Problem

- In 1900, mathematician David Hilbert delivered a now-famous address at the International Congress of Mathematicians in Paris.

- In his lecture, he identified 23 mathematical problems and posed them as a challenge for the coming century. The tenth problem on his list is of importance here.

# The Story of Hilbert’s Tenth Problem

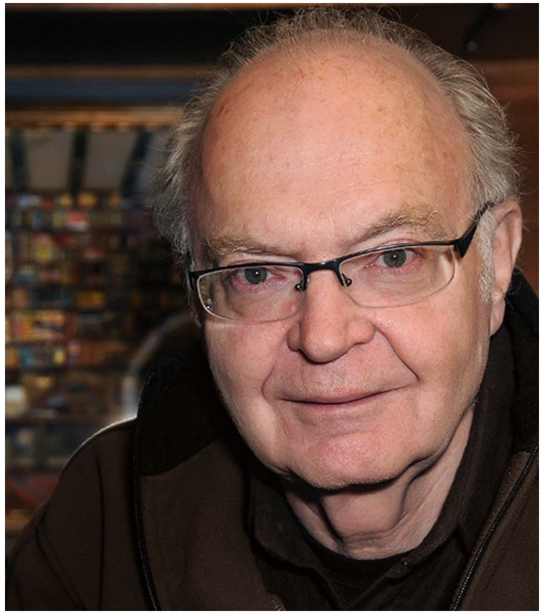
- Hilbert’s tenth problem was to devise an algorithm that tests whether a polynomial has an integral root or not.

- Interestingly, in the way he phrased this problem, Hilbert explicitly asked that an algorithm be “devised.”

- Thus he apparently assumed that such an algorithm must exist and someone need only find it. Today, we know that this assumption was indeed wrong!

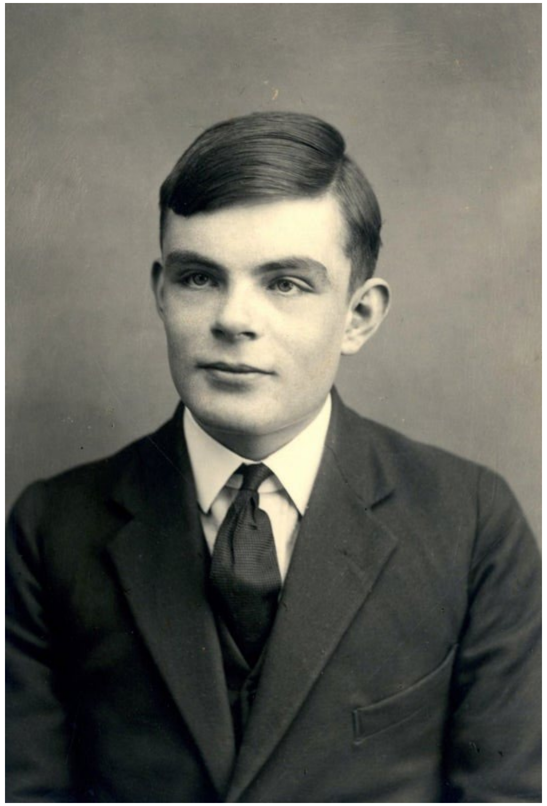
# Viewing Through the Lenses of History

# Viewing Through the Lenses of History



- Telling historical stories is the best way to teach. It's much easier to understand something if you know the threads it is connected to. […] The complete story is of many separate individuals building a magnificent edifice with a series of small steps.[2]

Donald Knuth



Alan Turing

Alonzo Church

Edmund Clarke

Michael O. Rabin

Stephen Cook

# Research: The Next Steps

# Research Avenues

- Compilers and Programming Languages
- Formal Verification and Synthesis
- Computational Complexity Theory
- Emergent Models of Computation

# The Millennium Problem

$$P \overset{?}{=} NP$$

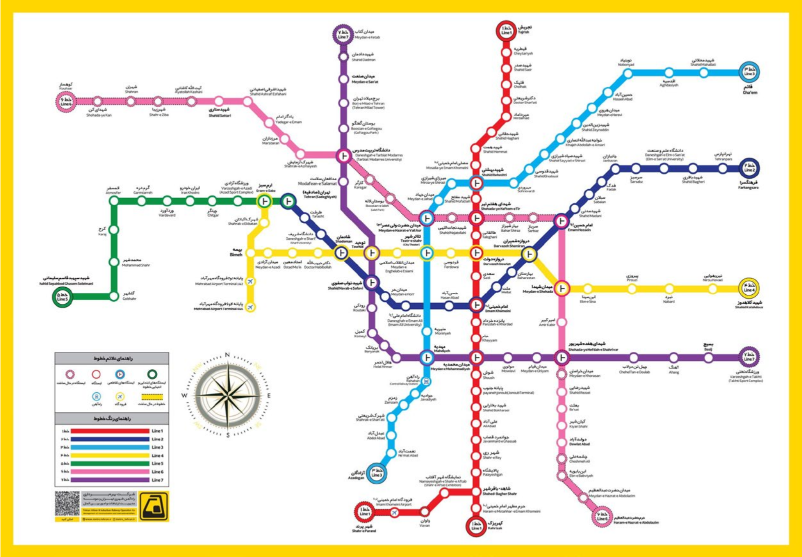
# Grand Challenge of the Century

- An opportunity for a grand challenge arises only rarely in the history of science. The construction and application of a verifying compiler that guarantees correctness of a program before running it [is of this kind].[3]

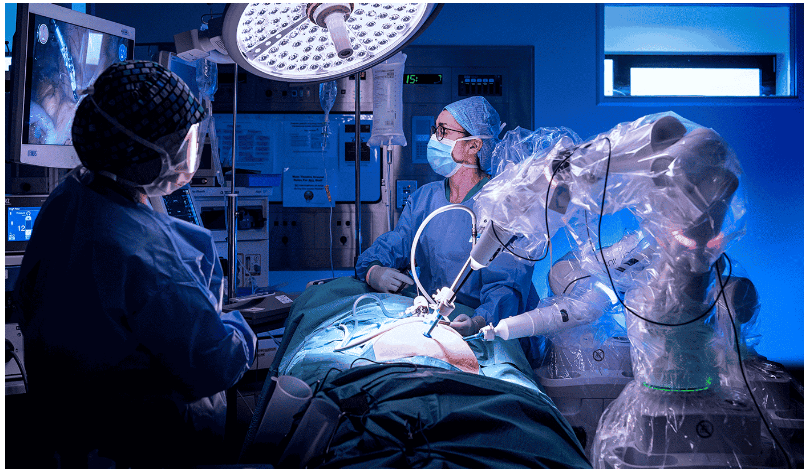
Tony Hoare

## Warehouse Management Systems

## Public Transportation Systems



## Medical Systems



# Trustworthy Intelligent Systems

- The growing impact of artificial intelligence systems has brought with it a set of risks and concerns such as errors and cyber-attacks and in general, use of these systems in safety-critical environments.

- Therefore, the question of verification and validation of AI systems, and, more broadly, of achieving trustworthy AI, has begun to demand the attention of the research community.

The automotive industry is a fine example of how these challenges can be obstructive and while the driver assistance systems are making their way to modern vehicles with intend to enable fully autonomous cars, there are still major concerns about the ability of the car to manage critical situations with minimum or no human interaction.

From: https://iaa.jhu.edu/emphasizing-social-benefits-could-improve-trust-in-autonomous-vehicles/

# The Suggested Solutions

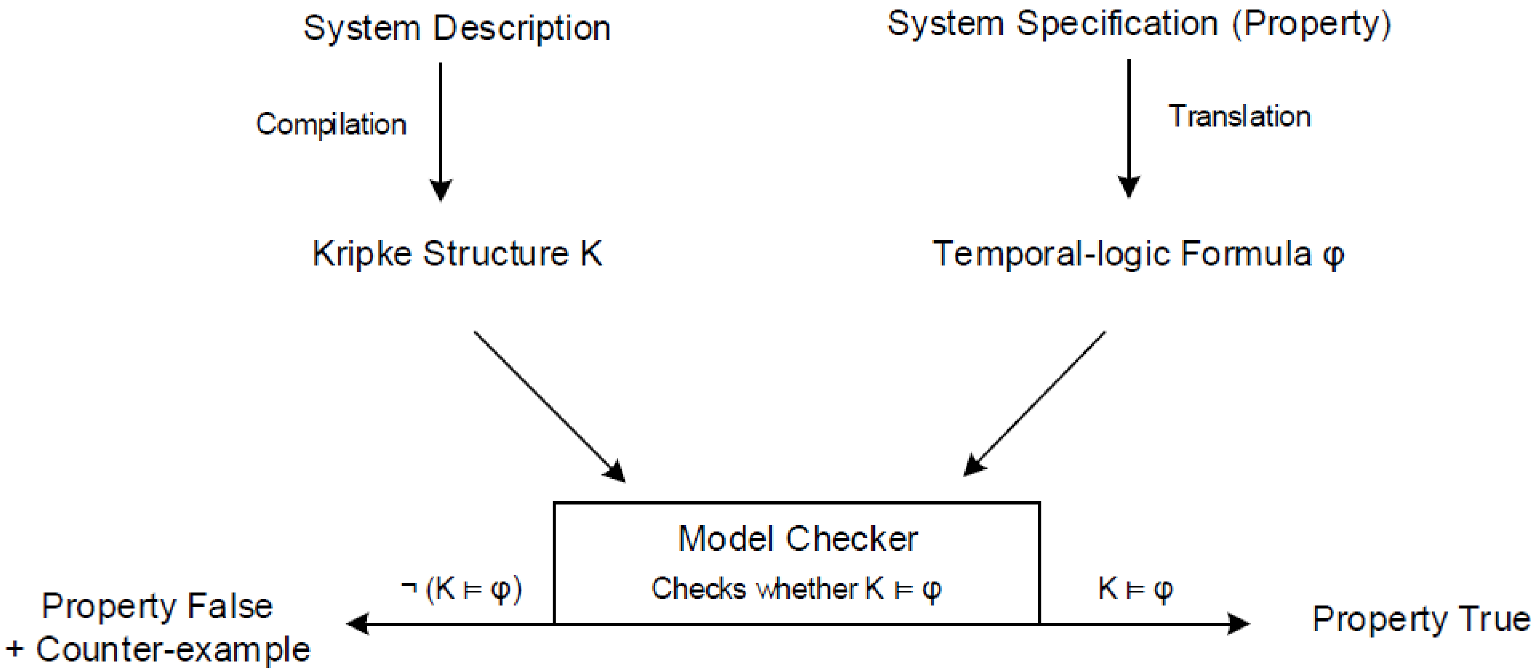
- To address these concerns, many approaches have emerged in the past few decades to supplement or even entirely replace the testing process and to increase the reliability of our systems.

- One of the most successful approaches is seeing computer systems as mathematical objects and making assertions about their intended behavior.

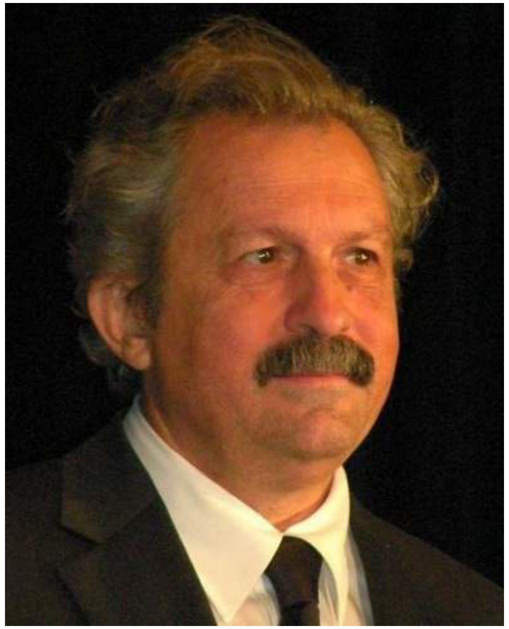
# Formal Verification in a Nutshell

- Formal verification – in simple terms - is the act of trying to establish the truth of these statements and guarantee the safe behavior of the system, at least to a extent that we can safely utilize them in the specified environment.
- Even though many contributions to this field in the past decades have led to great achievements and results, there are still many challenges in this field and it’s a subject of active research.

# Model Checking Technique



# In 2007, the inventors of Model Checking received the Turing Award

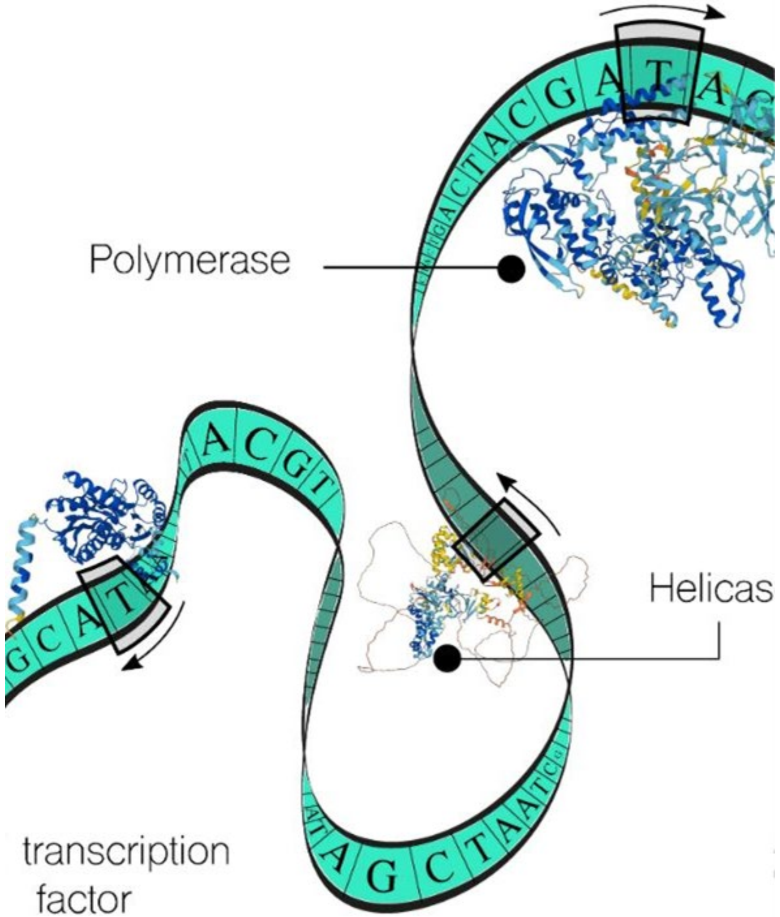


Joseph Sifakis

[4]

Some pioneers of formal methods gathered together on the sidelines of a conference.
From left to right: Amir Pnueli, Gerard Holzmann, Moshe Vardi, Bob Kurshan, David Dill, Ken McMillan, Edmund Clarke, Tom Henzinger, Limor Fix, Randy Bryant, Rajeev Alur, Allen Emerson.

# Emergent Models of Computation



[5]

- **Quantum Computation**
- **Biocomputation**
- **Photonic (Optical) Computation**
- **... ?!**

# Bibliography

1. Frege, Gottlob. “E. Heine's and J. Thomae's theories of irrational numbers.” The Philosophical Review 59, no. 1 (1950): 79-93.
2. Knuth, Donald E., and Len Shustek. “Let's not dumb down the history of computer science.” Communications of the ACM 64, no. 2 (2021): 33-35.
3. Hoare, Tony. “The verifying compiler: A grand challenge for computing research.” Journal of the ACM (JACM) 50, no. 1 (2003): 63-69.
4. Grumberg, Orna, and Helmut Veith, eds. “25 years of model checking: history, achievements, perspectives.” Vol. 5000. Springer Science & Business Media, 2008.
5. Goñi-Moreno, Ángel. “Biocomputation: Moving Beyond Turing with Living Cellular Computers.” Communications of the ACM 67, no. 6 (2024): 70-77.