Integrating Formal Methods and Large Language Models in Computer Engineering Education

Liliana Maria Favre
http://orcid.org/0000-0003-1370-1861
Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina & Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, Argentina

ABSTRACT

The development of safety- and mission-critical software has traditionally relied on formal methods to guarantee correctness, consistency, and verifiability. Despite their proven benefits, their use in mainstream software engineering remains limited due to the need for strong mathematical expertise, longer development times, rigid tools, and the absence of a large practitioner community. The emergence of artificial intelligence (AI) opens new possibilities to simplify formal development and encourages a revision of computer engineering curricula. This article proposes integrating the formal metamodeling language Nereus with AI throughout the software lifecycle. Current research explores how large language models (LLMs) such as ChatGPT can analyze, complete, and generate algebraic specifications, supporting syntactic and semantic reasoning, rewriting, and simulation. The article presents experiments redesigning an introductory algorithms and data structures course under this approach and introduces a metacognitive strategy to reinforce formal education in the AI era.

INTRODUCTION

The emergence of artificial intelligence calls for a comprehensive review of computer engineering curricula. Academic content must be carefully analyzed to establish clear priorities, update outdated material, and reorganize topics in a coherent and efficient manner. The goal is not to remove the foundational pillars of computer science such as programming, algorithms, and core software engineering principles, but rather to complement them with the knowledge and skills required to operate effectively in an ecosystem increasingly shaped by artificial intelligence. Within this context, certain areas gain relevance. These include the fundamental principles of artificial intelligence and machine learning, the

DOI: 10.4018/407444

mathematical foundations underpinning AI techniques, data science and data management, and software development practices enhanced by AI-supported tools. Equally important are topics related to ethics, regulation, governance, and responsibility in the design and deployment of AI systems. New content should be incorporated, such as prompt engineering and human-machine collaboration, requirements engineering and design with AI, cybersecurity with AI, and digital sustainability (energy impact of AI, green computing). In the IA era, characterized by a period of transition, software engineers and programmers bear responsibility for the code they generate with the assistance of AI tools. A competent auditor must possess a comprehensive understanding of the fundamental principles of the subject matter being audited. Consequently, the fundamental principles of computer science should be incorporated into the curriculum of basic computer science training.

Conversely, other content that was previously disregarded is now acquiring significance. For instance, while the merits of formal methods for developing software that is critical to safety and mission success are acknowledged, their application has not yet been extended to traditional programming. Their significant advantages are often overshadowed in mainstream software development by the perceived need for rigorous formalism, reliance on specialized expertise, longer development cycles, and the lack of a large and active practitioner community comparable to that of traditional programming. However, in the contemporary era, with AI capable of generating code in a matter of seconds, ensuring that the code meets specifications has become more important.

We consider that engineers should be instructed in formal methods to a greater extent than has been customary. This instruction should not be regarded as an isolated and theoretical subject; rather, it should be integrated into the entire development cycle in conjunction with artificial intelligence. The objective of this integration is to verify, validate, and build trust in systems. The integration of this system offers several advantages. On the one hand, artificial intelligence (AI) facilitates rapid prototyping; however, in the absence of a formal specification, guarantees are not provided. The use of artificial intelligence (AI) can significantly facilitate the drafting, revision, and translation of formal specifications. In particular, Large Language Models (LLMs) can be trained or guided to generate invariants, axioms, and even proofs derived from models and requirements (OpenAI, 2025), thereby supporting and accelerating formal development processes. This development has the potential to reduce the barriers to entry for formal specification, a process that has historically been regarded as "challenging." In certain areas, such as critical systems (e.g., aviation, healthcare, autonomous vehicles) and systems with embedded AI, it is imperative to verify that a model complies with specific security restrictions. Concurrently, emerging legal frameworks are promoting the need for the traceability and formal verifiability of intelligent systems.

The integration of AI with formal methods has been shown to yield both direct educational benefits and motivational advantages. The prospect of collaborating with an AI-powered assistant has garnered significant interest among students. A considerable number of individuals perceive artificial intelligence (AI) as an innovative, attractive, and cutting-edge field. This serves to pique their curiosity, inspiring an exploration of formal methods not only as a conventional technique, but also as a tool that facilitates their immersion in the captivating domain of AI. This experience engenders a sense of agency, empowering users to navigate and engage with emerging technologies. Consequently, it fosters intrinsic motivation and commitment to ongoing learning.

This chapter presents a proposal for integrating formal methods and artificial intelligence (AI) into foundational algorithms and data structures courses. In recent decades, a formal metamodeling language called Nereus, based on algebraic specifications, has been developed (Favre, 2009; Favre 2010; Favre, 2025). Nereus was introduced in the context of the definition of the Unified Modeling Language