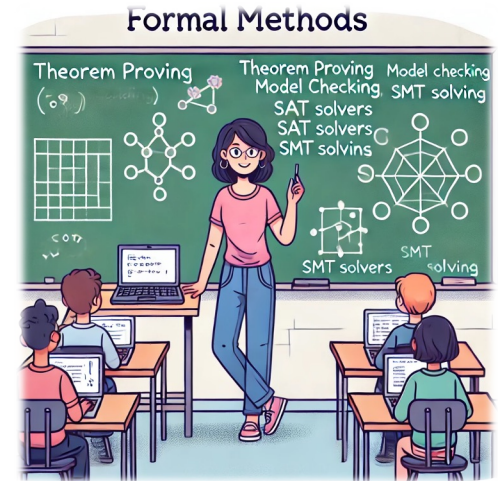


# Formal Methods Teaching Workshop

Emil Sekerinski, Leila Ribeiro

**Formal Methods Europe Teaching Committee**

[fme-teaching.github.io/](https://fme-teaching.github.io/)



- **FMTea workshop** on teaching Formal Methods: 2019, 2021, 2023, 2024
  - [fmtea.github.io/](https://fmtea.github.io/)
  - TFM 2004, TFM 2009
- **Collect Formal Methods courses** taught worldwide
- **Monthly tutorial series**
  - 21 speakers, next one on September 27, 2024
- **Work on a Curriculum of Formal Methods** in undergraduate/graduate Computer Science studies
- **Meeting on Wednesday**, during lunchtime, 12:50-14 (session 16, Room 3.0.1)
  - Devising FME guidelines for an FM Curriculum for undergraduate/graduate Computer Science studies
  - Working on a Book Series in Formal Methods

# Formal Methods Teaching Workshop

Preface of **TFM 2004, Ghent**, by Neville Dean Raymond T. Boute:

“Professional engineers can often be distinguished from other designers by the engineers’ ability to use mathematical models to describe and analyze their products.”

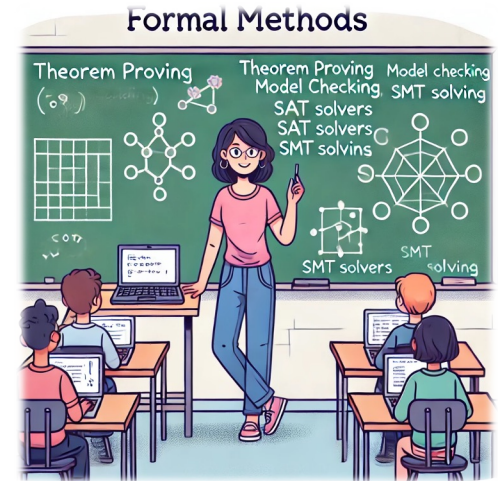
This observation by Parnas describes the de facto professional standards in all classical engineering disciplines (civil, mechanical, electrical, etc.). Unfortunately, it is in sharp contrast with current (industrial) practice in software design, where mathematical models are hardly used at all, even by those who, ...“aspire to be engineers.” The rare exceptions are certain critical applications, where mathematical techniques are used under the general name formal methods.

Yet, the same characteristics that make formal methods a necessity in critical applications make them also advantageous in everyday software design at various levels from design efficiency to software quality.

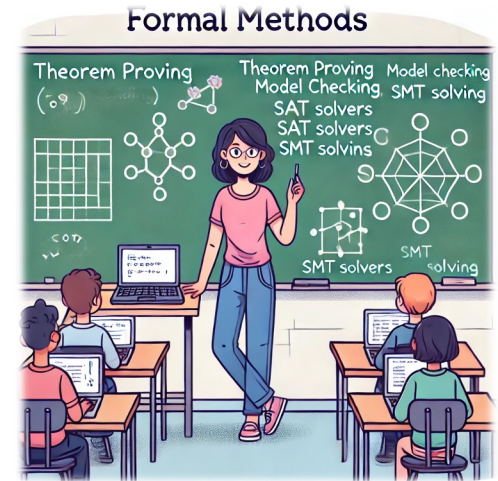
Why, then, is education failing with respect to formal methods?

- failing to convince students, academics and practitioners alike that formal methods are truly pragmatic;
- failing to overcome a phobia of formality and mathematics;
- failing to provide students with the basic skills and understanding required to adopt a more mathematical and logical approach to software development.

Until education takes these failings seriously, formal methods will be an obscure byway in software engineering, which in turn will remain severely impoverished as a result.



# Formal Methods Teaching Workshop



Session Chair: Emil Sekerinski

9:00 - 9:10 Opening Remarks

9:10 - 9:40 Soaibuzzaman, Jan Oliver Ringert. [Introducing GitHub Classroom into a Formal Methods Module](#)

9:40 - 10:10 Luca Padalino, Francesca Pia Panaccione, Francesco Santambrogio, Elisabetta Di Nitto, Matteo Rossi. [An Educational Module for Temporal Features in Alloy 6](#)

10:10 - 10:40 Gustavo Carvalho. [Teaching formal methods for 10 years: reflections on theories, tools, materials, and communities](#)

10:40 - 11:00 Break

Session Chair: Luigia Petre

11:00 - 12:00 Invited Talk: André Platzer. [The Significance of Symbolic Logic for Scientific Education](#)

12:00 - 12:30 Luca Negrini, Vincenzo Arceri, Luca Olivieri, Agostino Cortesi and Pietro Ferrara. [Teaching through Practice: Advanced Static Analysis with LiSA](#)

12:30 - 14:00 Lunch Break

Session Chair: Catherine Dubois

14:00 - 14:30 Dominique Méry. [Checking contracts in Event-B](#)

14:30 - 15:00 Stefan Hallerstede, John Hatcliff, Robby. Teaching with Logika: [Conceiving and Constructing Correct Software](#)

15:00 - 15:30 Achim D. Brucker, Diego Marmosler. [Teaching Formal Methods in Application Domains: A Case Study in Computer and Network Security](#)

# Formal Methods Teaching Workshop – Expo 16:00 - 17:30

John Hatcliff, Stefan Hallerstede:

1. Teaching with Logika: a safety-critical subset of Scala
2. End-to-end formal-methods-integrated development with HAMR: a model-based development framework for AADL and SysMLv2

Gustavo Carvalho: a unified overview of FMs with B

3. Specification (Atelier B)
4. Animation (ProB)
5. Visualisation (BMotionWeb)
6. Verification, refinement, and code generation (Atelier B)

Soaibuzzaman, Jan Oliver Ringert:

7. FM4SE Exercises with Autograding on GitHub Classrooms
8. Formal Methods Playground

