

# Teaching formal methods in academia: A systematic literature review

Rustam Zhumagambetov<sup>1</sup>[0000–0002–8061–4904]

Nazarbayev University, Nur-Sultan 010000, Kazakhstan  
`rustam.zhumagambetov@nu.edu.kz`

**Abstract.** The amount of literature on teaching formal methods has been growing. However, there is a lack of attempts to systematically review existing practices. This study attempts to identify challenges related to teaching formal methods by examining the literature on this topic. The literature review categorizes and systematizes the existing experience of teaching formal methods to students at universities. It presents obstacles reported as well as the strategies to deal with them that are expected to help current practitioners.

**Keywords:** formal methods · education · programming · teaching

## 1 Introduction

Current practices of developing software include its testing. However, from a formal point of view, it is not feasible to cover every possible piece of software functionality with tests. Especially this concern relates to safety or mission critical software, such as software for medical machines and satellites.

Formal methods in software engineering address this problem by utilizing mathematical frameworks of reasoning about programs, which is hard to understand for the majority of students. Courses that involve formal methods become the most unpopular.

To the best knowledge of the author there exists no systematic literature review that addresses challenges of teaching formal methods. One of the notable works that addresses challenges of teaching formal methods is the survey by Spichkova and Zamansky[24]. However, this survey is not comprehensive as it lacks the papers from the conferences on teaching formal methods. Thus, there is a need for systematic synthesis of the literature on the practices of teaching formal methods.

Despite the importance of formal methods for testing and verification of critical software, universities fail in teaching this aspect of computer science to students. This systematic literature review will categorize and systematize the existing experience of teaching formal methods to students at the university level.

## 2 Review Questions

To carry out this systematic literature review a number of activities, including database search, quality assessment as well as data extraction and synthesis, were conducted to formulate a conclusion.

The objective of this study is to identify the struggles that educators encounter during teaching formal methods in the higher education. A particular interest is strategies that used to overcome the struggles.

So, to achieve the objective the following research questions were formulated:

- *RQ1*. What are the challenges of teaching formal methods in the universities?
- *RQ2*. What are the strategies used to deal with these challenges?

## 3 Review Methods

According to Kitchenham and Charters [11], the following review components should be present to establish replicability of the review:

- Study search strategy
- Study selection criteria
- Study quality assessment
- Data extraction strategy

### 3.1 Study search strategy

Based on the objective of the paper, the following search query was used for the search: "teaching" AND "formal methods". The terms were matched to titles, abstracts and keywords.

The following databases were used for search:

- IEEEExplore (<https://ieeexplore.ieee.org/>)
- ACM Digital library (<https://dl.acm.org/>)
- Elsevier Science Direct (<https://www.sciencedirect.com/>)

The author have decided to include papers from conference proceedings. Proceedings of the following conferences were included:

- TFM 2009
- FMET 2008
- TFM 2006
- FMED 2006

As can be seen in Figure 1, the primary search have found 151 studies.

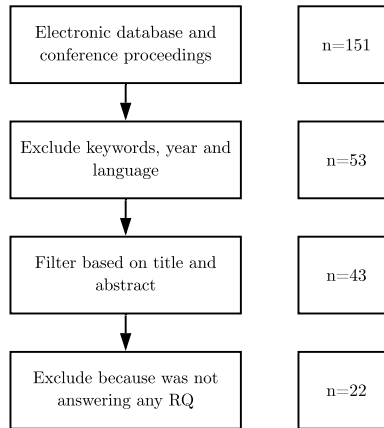


Fig. 1. A flowchart that represents research protocol

### 3.2 Study selection criteria

After the initial review of the queries, some keywords were identified as irrelevant to the study. Here is the list of excluded keywords: statistic, statistics, Grammar, Second Language Learning, formal methods of estimation, Economics education, Law students, Physical education, Physicians, Emergency Medicine Education. Other criteria for selection are papers should be published not later than 2001 and be written in English language. Number of studies at this stage is 53.

After keywords exclusion, all papers were reviewed by the author to perform third stage of selection. During this stage we filtered papers based on their relevance to the research questions.

Then, papers were filtered based on title and abstract. During this filtering process studies that claimed to teach formal methods outside of universities, presented formal methods in other contexts were excluded. For example, title of [25] states about experience in middle school and [7] about formal methods in electronic government. At this stage the number of papers was reduced to 43

### 3.3 Study quality assessment

In this stage rest of the papers were read and assessed on how they answer at least one of the research questions. At this stage, primary targets for filtering out were papers that have not neither evidence nor opinion regarding how challenges of teaching formal methods or what strategies were used improve experience of the students that participate in the course. Examples of the filtered out papers are [18] and [14]. The former paper mostly consists of anecdotes that are aimed to motivate anyone, who reads them, to use formal methods. The latter paper is the one, which focused more on the tool, rather than how the tool solves

the problem of students. The final number of papers used to answer research questions is 22.

### 3.4 Data extraction strategy

The resulting set of the papers is a mixture of teaching experience that was recorded either through a session or within a certain period, when course was offered. Other papers demonstrate the use of software or teaching approaches to solve general or a particular problem along with student’s feedback.

Due to the nature of the ”raw” material, structured data extraction becomes either very difficult or impossible to conduct in the objective manner. So the author used a word processor to identify some of the themes presented in the studies, grouping them as necessary.

## 4 Results

In this section we report findings about research questions. The purpose is to provide themes discovered in the studies.

### 4.1 RQ1 - What Are the Challenges of Teaching Formal Methods in the Universities?

We found 17 studies that identified 3 main themes that answer this question: students scepticism, difficulties with auxiliary software and challenges with materials.

**Students’ Scepticism.** This is one of the biggest challenges identified by majority of researchers. Throughout the literature there were two prominent problems that are related: student’s scepticism about formal methods as well as isolation of formal methods from application in other courses.

*Students Are Skeptical about the Usefulness of Formal Methods.* This is the major problem identified in 8 studies [2, 3, 4, 6, 9, 17, 19, 26]. Blanco et al. [2] discusses how program verification is viewed by students as a ”an additional burden”. Brakman et al. [4] perceives that students see little practicality in use of formal methods.

*Isolation of Formal Methods From Application in Other Courses.* Boute [3] observes often such behavior is motivated by little exposure to formal methods across curriculum. Supporting, Sekerinski [22], reporting his experience of teaching a course on software design with formal methods, identifies an isolated use of formal methods as a main obstacle that prevents its flourishing.

**Difficulties with Auxiliary Software** Another theme that was identified throughout review is difficulties with tools that are used to teach formal methods. Steep learning curve and bad user experience are major challenges that support the theme.

*Steep Learning Curve for Tools.* Lipaczewski and Ortmeier [15] recognized a steep learning curve as one of the failures of tools that use formal methods. Schreiner [21] supports them adding that teaching time is too scarce to consider teaching how to use this "difficult to learn and/or inconvenient to use" software.

*Tools Give Little Feedback.* In their course Bayley, Lightfoot and Martin [1] avoid using software tools, because tools can "produce a fairly cryptic and discouraging response". Gopalakrishnan [9] observed that most tools lack a good user experience and failing do not provide insight at why they fail.

**Challenges with Course Materials.** The third theme that was common to a number of studies is challenges with course materials. Such challenges include dryness of the course, diversity of materials, lack of comprehensive textbook and course require students to have better mathematical knowledge than they have.

*Dryness of the Course.* Two studies, [4] and [19] identified dryness of the course as a problem of teaching formal methods. In particular, Brakman et al. [4] reports that "lack of visualization" is the primary reason of ineffectiveness of teaching formal methods.

*Diversity of Materials.* Gopalakrishnan [9] notes that there are a variety of tools that use formal methods ; however, there are few initiatives to adopt them. These tools "face the uphill battle of vying for attention and commitment from teachers" [9].

*No Comprehensive Textbook.* While there exists a diversity of materials, some authors find it difficult to navigate through them. In particular, Kofroň, Parížek and Šerý [12] report that they were not satisfied with comprehensiveness of one of the textbooks used for one of their courses.

*Inadequate Background in Mathematics.* Several authors [8, 13, 16, 23] report students that take courses in formal methods often do not know a certain concepts that are used in the course. Consequently, such inadequacy requires inclusion of additional materials into the course.

#### 4.2 RQ2 - What Are the Strategies Used to Deal with These Challenges?

We have found 15 papers and 4 major themes that answer this question that are formulated as a form of advice: practice, engage students, avoid isolation of formal methods and simplify material.

**Practice.** This advice was formulated because several authors attributed the success of the formal methods courses they ran to the extensive practice. They provided a large number of examples to the students to facilitate the ingestion of theoretical and practical components of formal methods.

*Use a Lot of Examples.* Aceto et al. [6] report that they avoid presenting general theory, preferring investigation of particular instances. They declare that examples play "primary role" in their practice.

*Real-life Examples and Projects.* Some practitioners [4, 5, 20] make stronger claims by saying that examples should involve "real-life", not artificial instances of the problem. For instance, Brakman et al. [4] discusses how presenting verification of Bluetooth protocol as a course project helped their students to increase understanding of formal methods.

**Engage Students.** This strategy was devised to describe several reported practices that increased engagement and motivation of the students with formal methods.

*Gamify.* Prasetya et al. [19] presents a game, FormalZ, that presents concepts of formal methods in fun and engaging manner. It also introduces a competitive environment by introducing a leaderboard, a ranking list. Their preliminary results show positive feedback from students.

*Use Electronic Voting systems.* Miller and Cutts [16] introduce an electronic voting system (EVS) to help students to actively participate. EVS often consists of a several buttons that represent answer choices and a display that presents a question. Often EVS are anonymous and are instruments to increase student engagement rather than test of the knowledge. Miller and Cutts [16] report use of EVS increased confidence of student to "to answer questions within the class".

*Use Formal Methods in Programming Contests.* Shilov and Yi [23] propose to engage students with theory through organization of programming contests. They notice that training sessions the precede the contests are an excellent opportunity to engage students with theory.

**Avoid Isolation of Formal Methods.** This direction came from several studies that report an underrepresentation of the formal methods across curricula. Such situation leads to formal methods looking esoteric, decreasing student's motivation [3, 22].

*Start Early in Introductory Course.* Blanco et al. [2] reports that the course on introductory programming that was based on formal methods was helpful for students to learn appreciation of formal methods. They report that the course had lasting effect based on "the optional courses they choose later on in the programmes, and the topics they choose for their final projects" [2].

*Use a Combination of Formal Techniques.* Some educators claim that they achieve best results when they explain several formal techniques. Hallerstede and Leuschel [10] utilize combination of "formal proof, model-checking, and animation" to improve accuracy of a formal model.

*Juxtapose of Taught Formal Methods with Other Methodologies.* To overcome students' scepticism about formal methods Catano and Rueda [6] ask students throughout the course to reflect upon pros and cons of the formal methods compared to the already learned traditional approaches.

**Simplify Material.** This strategy is intended to simplify not formal methods, but the material used to deliver the lesson. While it is difficult, or impossible to make formal methods accessible for everyone, some educators have found ways of making their delivery friendlier to the listeners.

*Use Another, Simpler Tool.* Some studies [15, 21] suggest that to overcome challenges associated with software, which uses formal methods, it is the best to use other, simpler tools. For example, Lipaczewski and Ortmeier [15] propose SAML with its plugins for IDE and web-based user interface. SAML is intended as tool for building models that are more friendly to the user than existing alternatives.

*Use Common Languages that Have Built-in Expressivity.* Another study avoids specialized formal methods software and languages. Instead, Gopalakrishnan [9] uses a general purpose language, like Python, for "Models of Computation" course. It has features, like list comprehension that helps writing definitions with mathematical notation.

*Use Less Formal Testing as a Medium for Teaching Some of the Formal Methods.* Utting and Reeves [26] suggest that interweaving of traditional testing approach and elements of formal methods helped their students to have a better satisfaction with using formal methods.

## 5 Discussion

This systematic review have demonstrated that there is a number of challenges, like students' scepticism, difficulties with computer-aided assistants and challenges with the course materials. There are a few practices that can help to overcome these challenges, like use of examples, gamification, teaching of combination of formal methods and simplification of the delivery. The author believe that some of the discovered challenges are valid and hope that provided strategies will help to overcome them.

## 5.1 Limitations

Even though this literature review attempts to be objective, there is no ideal, objective criteria that would quantify success or failure of the provided solutions. Most of the researchers relied on the students feedback, course enrollment number or their own senses to draw a conclusion about usefulness of certain techniques.

## 6 Conclusion

A systematic literature review was performed on 22 papers that report practices of teaching formal methods in universities. The aim of this review was to identify the struggles that educators encounter while teaching formal methods, as well as exploring strategies that are used to deal with these struggles. The findings allowed to formulate 8 themes of challenges that were further grouped into 3. We have found 11 themes of strategies that were grouped into 4.

## References

1. Bayley, I., Lightfoot, D., Martin, C.: Teaching the Oxford Brookes Formal Specification Module. *Teaching Formal Methods* p. 5 (2006)
2. Blanco, J., Losano, L., Aguirre, N., Novaira, M.M., Permigiani, S., Scilingo, G.: An Introductory Course on Programming Based on Formal Specification and Program Calculation. *SIGCSE Bull.* **41**(2), 31–37 (Jun 2009). <https://doi.org/10.1145/1595453.1595459>
3. Boute, R.: Teaching and Practicing Computer Science at the University Level. *SIGCSE Bull.* **41**(2), 24–30 (Jun 2009). <https://doi.org/10.1145/1595453.1595458>
4. Brakman, H., Driessen, V., Kavuma, J., Bijvank, L.N., Vermolen, S.: Supporting Formal Method Teaching with Real-Life Protocols. *Formal Methods in the Teaching Lab: Examples, Cases, Assignments and Projects Enhancing Formal Methods Education* pp. 59–68 (2006)
5. Catano, N.: An Empirical Study on Teaching Formal Methods to Millennials. In: 2017 IEEE/ACM 1st International Workshop on Software Engineering Curricula for Millennials (SECM). pp. 3–8. IEEE, Buenos Aires, Argentina (May 2017). <https://doi.org/10.1109/SECM.2017.1>
6. Catano, N., Rueda, C.: Teaching Formal Methods for the Unconquered Territory. In: *Teaching Formal Methods*, vol. 5846, pp. 2–19. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-04912-5\\_2](https://doi.org/10.1007/978-3-642-04912-5_2)
7. Davies, J., Gibbons, J.: Formal methods for electronic government. In: *Formal Methods Education and Training*, pp. 74–79 (2008)
8. Feinerer, I., Salzer, G.: Automated Tools for Teaching Formal Software Verification. *Teaching Formal Methods* p. 5 (2006)
9. Gopalakrishnan, G.: Formal Methods for Surviving the Jungle of Heterogeneous Parallelism. In: 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum. pp. 1321–1324. IEEE, Shanghai, China (May 2012). <https://doi.org/10.1109/IPDPSW.2012.164>



10. Hallerstede, S., Leuschel, M.: How to Explain Mistakes. In: *Teaching Formal Methods*, vol. 5846, pp. 105–124. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-04912-5\\_8](https://doi.org/10.1007/978-3-642-04912-5_8)
11. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering (2007)
12. Kofroň, J., Parížek, P., Šerý, O.: On Teaching Formal Methods: Behavior Models and Code Analysis. In: *Teaching Formal Methods*. pp. 144–157. Springer Berlin Heidelberg (2009)
13. Loukanova, R.: Teaching Formal Methods for Computational Linguistics at Uppsala University. *Teaching Formal Methods* p. 6 (2006)
14. Méry, D.: A simple refinement-based method for constructing algorithms. *SIGCSE Bull.* **41**(2), 51–59 (Jun 2009). <https://doi.org/10.1145/1595453.1595462>, <http://doi.acm.org/10.1145/1595453.1595462>
15. Michael Lipaczewski, Frank Ortmeier: Teaching and Training Formal Methods for Safety Critical Systems. In: *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*. pp. 408–413. IEEE, Santander, Spain (Sep 2013). <https://doi.org/10.1109/SEAA.2013.54>
16. Miller, A., Cutts, Q.: The use of an electronic voting system in a formal methods course. In: *Formal Methods in the Teaching Lab: Examples, Cases, Assignments and Projects Enhancing Formal Methods Education. A Workshop at the Formal Methods 2006 Symposium*, Hamilton, Ontario, Canada, 26 Aug 2006, pp. 3–8. McMaster University, Hamilton, ON, Canada (2006)
17. Ölveczky, P.C.: Teaching Formal Methods Based on Rewriting Logic and Maude. In: *Teaching Formal Methods*, vol. 5846, pp. 20–38. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-04912-5\\_3](https://doi.org/10.1007/978-3-642-04912-5_3)
18. Parnas, D.L., Eng, P., Soltys, M.: Basic science for software developers (2006)
19. Prasetya, W., Leek, C., Melkonian, O., ten Tusscher, J., van Bergen, J., Everink, J., van der Klis, T., Meijerink, R., Oosenbrug, R., Oostveen, J., van den Pol, T., van Zon, W.: Having Fun in Learning Formal Specifications. In: *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. pp. 192–196. IEEE, Montreal, QC, Canada (May 2019). <https://doi.org/10.1109/ICSE-SEET.2019.00028>
20. Roychoudhury, A.: Introducing model checking to undergraduates. In: *Formal Methods Education Workshop*. pp. 9–15 (2006)
21. Schreiner, W.: The RISC ProofNavigator: a proving assistant for program verification in the classroom. *Formal Aspects of Computing* **21**(3), 277–291 (May 2009). <https://doi.org/10.1007/s00165-008-0069-4>
22. Sekerinski, E.: Teaching the Mathematics of Software Design (2006)
23. Shilov, N.V., Kwangkeun Yi: Engaging students with theory through ACM collegiate programming contests. *Communications of the ACM* **45**(9), 98–101 (Sep 2002)
24. Spichkova, M., Zamansky, A.: Teaching of Formal Methods for Software Engineering. In: *Proceedings of the 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering*. pp. 370–376. SCITEPRESS - Science and Technology Publications, Rome, Italy (2016). <https://doi.org/10.5220/0005928503700376>
25. Spies, K., Schätz, B.: A playful approach to formal models a field report on teaching modeling fundamentals at middle school. In: *Formal Methods in the Teaching Lab Examples, Cases, Assignments and Projects Enhancing Formal Methods Education*, pp. 45–53 (2006)

26. Utting, M., Reeves, S.: Teaching formal methods lite via testing. *Software Testing, Verification and Reliability* **11**(3), 181–195 (Sep 2001). <https://doi.org/10.1002/stvr.223>