Teaching Statement

Adithya Murali
University of Illinois Urbana-Champaign
https://muraliadithya.github.io

My teaching philosophy aims to enable the individual intellectual development of students, primarily using approaches characteristic of peer instruction. I learn about students' professional goals and tailor classroom experiences accordingly. **I view myself as a mentor and collaborator in the student's intellectual journey and derive value from being a part of their transformation.** I use frameworks from project management to design the learning process and techniques from stage acting to structure communication.

My approach is based on my teaching experiences spanning a decade. I found that students learned most effectively when they considered me a peer. A peer is approachable, available, patient, relatable, and collaborative. When working with a peer students can acquire their desired level of proficiency without fearing judgment of their intelligence or competence. My methods ensure that students experience the same curiosity, enthusiasm, and comfort that they experience with peer learning[1].

Teaching Experience

My teaching experience spans several formal and informal roles that I exercised through my undergraduate and graduate studies. I describe below a few vignettes from my experiences that illustrate my teaching philosophy.

Undergraduate Experience As an undergraduate, I conducted weekly problem-solving sessions for a course on linear algebra and complex analysis. During this time, I informally experimented with different staging and communication models. I converged on a partially flipped learning model that is less collaborative and more centered around the teacher than a traditional flipped learning model. I queued the students' questions at the beginning of the class and spent a fixed amount of time on each element of the queue, communicating primarily with a subset of the students while the others listened and learned from their peers' exchanges. I staged the class in the usual way standing in front of a board while facing the students, allowing every student in the class the choice of when to join or leave the individual discussions. This can be likened to a traditional flipped learning model where students deliver mini-lectures, adapted to the idea of students leading discussions on specific problems/concepts and actively soliciting the teacher for specific pieces missing from their understanding.

This approach was very successful, and I observed that **in-class participation was higher** for my sessions compared to the other instructional-style tutorials. This was encouraging since tutorial sessions were mostly populated by students who were struggling with the concepts and could not apply the ideas discussed in the lecture on their own. I also determined anecdotally that students who attended my classes understood the material more deeply and required less practice to acquire proficiency with each problem type.

Graduate Experience I have been a Teaching Assistant for three courses (a total of five appointments) during my Ph.D. As a TA for an undergraduate course on *Programming Languages and Compilers*, I continued to experiment with **methods for incorporating students' individual needs and abilities into the learning process**. One such opportunity occurred when many students were struggling with formal methods topics such as regular expressions and context-free grammars. I analyzed the problem and observed that these were students in their junior/senior years who had strong affinities with programming-oriented methods of learning. To help them learn these topics I leveraged their strengths by rephrasing the difficult problems into programming tasks. For example, I corresponded the Kleene star to a certain kind of loop. **By crafting intuitive abstractions, I helped students relate the material to their own experiences.** This had very good results, and the struggling students could now solve the problems easily. I shared these ideas with other TAs who observed similar results with their students.

My most recent teaching experience has been as a TA for two graduate courses on *Logic in Computer Science* and *Trustworthy AI Systems*. During these appointments, **I undertook mentorship on how to develop a course**

[1] I discuss issues relating to inclusive evaluation methodologies under individualized teaching models in my DEI statement.

from the instructors by participating in course design discussions encompassing curriculum, learning objectives, lesson plans, and assessments. These appointments also offered me a lot of autonomy: I conducted office hours, created and graded the assignments and exams, formulated project topics for course projects, and mentored the students through the term on their projects. **A particular strength I brought to my role was my research experience with industry-grade logic reasoning engines (SMT solvers)**. Instruction in logic can often be very theory-centric. However, I used my experience with SMT solvers to create assignments with dual theoretical and application components. I structured assignments by explicitly stating the learning objectives for each problem and connected them to the material covered in the lectures. This helped students approach their learning from a practical standpoint and enabled them to test their conceptual understanding easily. Owing to the success of these assignments, I have secured a special TA appointment where **I will be working on developing a full suite of auto-graded assignments** for these courses. I am excited to bring this expertise to future teaching roles.

Finally, I have delivered guest lectures in courses on logic, formal methods, and programming languages based on my research expertise. During these lectures, **I connect concepts covered during the course to research problems**. My talks have been well-received, with one of them leading to a productive research collaboration with an undergraduate student [5] and another leading to a tutorial at POPL 2024 [1].

Mentorship

Mentorship offers me another channel for exercising my teaching philosophy. Compared to teaching, the objectives are larger in scope and the mentee has a much greater influence in defining the goals of the relationship. Since mentorship often involves "unknown unknowns" from the mentee's perspective, I augment my emphasis on individual development with a **transformational leadership** style[2]. I engage in a variety of mentorship efforts, including reviewing graduate school applications, research mentorship, academic advising, and mental health support. I discuss research mentorship below and detail my efforts on other aspects in the DEI statement.

I have mentored 13 undergraduate and graduate students on research projects and co-authored 5 papers [2, 3, 4, 5, 6] with 6 mentees. Five of my mentees had their first research experience with me, and many are currently pursuing Ph.D. programs at top universities like UT Austin and CMU. One of the most rewarding moments in my experience was when one mentee chose to pursue a Ph.D. as a direct consequence of the joy they experienced working with me, and chose to work on directions related to our collaboration! I cultivate long-term relationships with my mentees and continue to advise them through the stages of their development as independent researchers.

Apart from this, I regularly advise students on soft skills in research such as writing and presentation skills. I have been **organizing the UIUC Formal Methods seminar** over many semesters where several students have given their first talks. I also use the FM seminar to teach junior Ph.D. students how to engage with research outside their expertise to develop their research tastes and cultivate a wider perspective of the field. I find that **mentees gain a more holistic view of scholarship** as a consequence of these efforts.

Teaching Interests

At the undergraduate level, I am interested in teaching courses on Programming Languages, Compilers, Software Engineering, Logic, Theory of Computation, Discrete Mathematics, and Data Structures and Algorithms. I would also be able to teach other undergraduate courses with some preparation. At the graduate level, apart from courses on formal methods, programming languages, and software engineering, I am especially interested in teaching courses on **software verification and logic**.

A key initiative I intend to work on is **creating hands-on assignments that introduce students to automated logic engines**. This will help them acquire practical problem-solving skills based on a deep theoretical understanding. These assignments would also be **automatically graded**, which can be especially useful for undergraduate courses such as discrete mathematics. As I have indicated above, I am currently working on developing such auto-graded assignments at UIUC and I can transfer this expertise to other courses at my new institution.

Finally, I am also interested in teaching special topics courses or research seminar courses on topics relating to my research such as automated reasoning, program synthesis, and logic learning. The **intersection of programming languages and machine learning** is an emerging area, and I plan to develop a course in collaboration with machine learning faculty that covers topics in this space like neurosymbolic programming, machine learning for theorem proving, and trustworthy AI.

[2] A transformational reader leads by inspiring their team to work towards a shared vision, challenging them to be better professionals and developing their capacity via individual support, coaching, and mentoring. For a detailed discussion, see a standard text [7].

References

[1] **Adithya Murali** and Madhusudan Parthasarathy. *Automated Datastructure Verification using Unfoldings and SMT Solving: Foundations and FO-Completeness (POPL 2024 - TutorialFest) - POPL 2024*. publisher: POPL 2024. URL: https://popl24.sigplan.org/details/POPL-2024-tutorialfest/10/-Automated-Datastructure-Verification-using-Unfoldings-and-SMT-Solving-Foundations-a (visited on 11/24/2023).

[2] **Adithya Murali**, Cody Rivera, and P. Madhusudan. "Predictable Verification using Intrinsic Definitions of Datastructures". In: *Under Submission* (2023).

[3] **Adithya Murali**, Lucas Peña, Eion Blanchard, Christof Löding, and P. Madhusudan. "Model-Guided Synthesis of Inductive Lemmas for FOL with Least Fixpoints". In: *Proc. ACM Program. Lang.* 6.OOPSLA (Oct. 2022). DOI: 10.1145/3563354. URL: https://doi.org/10.1145/3563354.

[4] Paul Krogmeier, Zhengyao Lin, **Adithya Murali**, and P.Madhusudan. "Synthesizing Axiomatizations Using Logic Learning". In: *Proc. ACM Program. Lang.* 6.OOPSLA (Oct. 2022). DOI: 10.1145/3563348. URL: https://doi.org/10.1145/3563348.

[5] **Adithya Murali**, Atharva Sehgal, Paul Krogmeier, and P. Madhusudan. "Composing Neural Learning and Symbolic Reasoning with an Application to Visual Discrimination". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. Ed. by Luc De Raedt. Main Track. International Joint Conferences on Artificial Intelligence Organization, July 2022, pp. 3358–3365. DOI: 10.24963/ijcai.2022/466. URL: https://doi.org/10.24963/ijcai.2022/466.

[6] **Adithya Murali**, Hrishikesh Balakrishnan, Aaron Councilman, and P. Madhusudan. "Automating Program Verification for Frame Logic". In: *Under Submission* (2023).

[7] B.M. Bass and R.E. Riggio. *Transformational Leadership*. Taylor & Francis, 2006. ISBN: 9781135618889. URL: https://books.google.com/books?id=2WsJSw6wa6cC.