TEACHING STATEMENT SAÚL A. BLANCO NOVEMBER 13, 2019

In my time with the CS Department, I have taught the undergraduate intro to AI class (B351), the discrete structures class (C241), and currently a graduate class on combinatorics and computing. Before joining the CS department, I taught other CS and Info courses (H241: honors discrete structures, C290: topics in computing, games and puzzles, I308: information representation, and I231/C231: intro to math for cybersecurity). Furthermore, I have two papers, one single-authored and one co-authored with an undergraduate instructor (UI) for my intro to AI class, in the SIGCSE conference, which is the largest national conference in CS education. These papers relate to active learning in discrete mathematics and the effects of automated vs. human feedback in the intro to AI class, since for our intro to AI class we use an in-house automatic grading system designed by our UIs after several semesters.

In the classroom, I incorporate both active and collaborative learning techniques while providing a safe environment in which students can work without fear of making mistakes.

I set aside time for the class to work on problems that motivate the day's topic before introducing any definitions or notation. For example, I ask students to solve knights-and-knaves type of puzzles before introducing formal logic notation, or the classic man crossing the river with a wolf, goat, and cabbage puzzle before talking about space states and search trees. When I ask someone to present their solutions, I often first hear comments like "I know the answer, but I don't know how to explain it." After a few hints, they start writing valid justifications and are happy to present them to their classmates, which sets a positive tone for the entire term. Other examples I use include the Königsberg bridge problem to introduce graph theory, the Collatz conjecture to introduce periodicity, and the Hanoi tower to introduce recursively defined structures. When students work through these motivating examples, they sometimes come to the realization that notation is needed, so the formal definitions seem more natural.

In addition to the motivating examples to introduce topics, I design worksheets with carefully chosen problems that illustrate the topics being covered. Students work on these problems during class and we discuss some of their solutions using the room's document camera. While some find this intimidating at first, when someone makes a mistake during their presentations, I use the opportunity to ask the rest of the class to correct and complete the answers. For example, when doing computations in Bayesian networks, it is easy for students with little exposure to the topic to confuse the process, and I frequently show their incomplete or even incorrect work and ask the rest of the class to help correct it under my supervision. This collaborative approach not only reduces fear of mistakes but also builds rapport, which is important as it helps students being comfortable asking questions. I also use team-based learning techniques when appropriate. In my discrete math course, I assign readings ahead of time and then quiz students, both individually and in semester-long teams. During the team portion of the quiz, I often observe students teaching themselves concepts that have yet to be covered. For instance, I see them drawing arrow diagrams to explain the definitions of one-to-one and onto, producing counterexamples to statements involving nested quantifiers, and talking about the element method to prove set identities before these concepts are discussed in class. This approach is very effective in showing students that they can learn by themselves, which builds soft skills such as communication skills, confidence, and the ability to work in teams. For higher-level courses, I include final projects, which are carried out in small teams, represent a key component of semester. Moreover, we have had public poster sessions where students present their projects. These presentations help the students prepare to make professional presentations and demos and provide visibility for our class. Indeed, other students get interested in taking our AI courses.

Another important feature in my teaching is the heavy involvement of undergraduate instructors as teaching assistants. These students are selected due to their performance in the class and usually stay with the class for a few semesters. Our School has had the tradition of successfully using UIs in several courses. There is plenty of evidence in education literature showcasing the benefits of using UIs: students relate better to them and feel more comfortable asking questions, the UIs get a better mastery of the subject and can showcase their work to future employers, and it is usually possible to hire several UIs in one class which allows for multiple office hours during the week. The UIs I've worked with have designed an in-house grading tool that we use for the intro to AI class and that can be used for other courses. Moreover, we have co-authored a paper relating to this grading tool that will be presented in next year's SIGCSE.

I've found that using these active and collaborative learning techniques transform students from passive learners to engaged individuals. Not everyone in my classes will be a mathematician or a computer scientist, but they will all use the reasoning skills and other soft skills that my classes help them develop.