

Programming can be incredibly empowering, but there are substantial cultural, institutional, and technical barriers that have limited women, members of minority groups, people from poor communities, and people with disabilities from participating fully in the programming community. I am committed to breaking down these barriers with my research, teaching, advising, service, and outreach. My approach involves, first and foremost, listening to and acting on the advice of people from these groups as well as professional organizations like the National Center for Women & Information Technology (NCWIT). Beyond that, I highlight some specific initiatives that I have taken, and that I plan to take, below.

1 Outreach

One way to increase the diversity of the broader programming community is to introduce programming into communities where there is more diverse participation today. I plan to build on connections that I have already developed in the biological sciences, the visual arts, and music to teach and to help others teach domain-specialized introductory programming workshops and courses using the tools that my research group is designing, alongside other appropriate tools and techniques.

Biological Sciences I was a graduate student in neurobiology before transferring to focus on programming languages and tools. After transferring, I continued to participate in the Graduate Training Program offered by the Center for the Neural Basis of Cognition at CMU. This involved attending and giving talks and participating in yearly retreats. I have also attended the Society for Neuroscience’s Annual Conference (which draws over 30,000 researchers) three times and the annual Cosyne meeting twice. Consequently, I have well-developed intuitions for the challenges that come up when introducing programming tools, which are of increasing importance in both experimental and theoretical biology, to students who have had substantial mathematics education but little to no formal programming education. I have also studied the literature in this area. One important finding is that building on pre-existing mathematical and domain knowledge helps with both general and minority retention in a *CS for Scientists* course [Dodds et al., SIGCSE 2008]. I design the programming tools that are the focus of my current and ongoing research—Hazel and typy—to take advantage of and reinforce students’ mathematical intuitions through a combination of functional programming techniques (where mathematical expressions, rather than machine-oriented commands, are central) and techniques that allow students to use notation, both textual and graphical, with which they are familiar from their other educational experiences. I am currently working with Prof. Ben Shapiro, who studies CS Education, and a graduate student, David Moon, to adapt the Bootstrap Data Science course module to use Hazel. When this is ready, we plan to teach summer modules for student scientists, taking special care to recruit a diverse pool of students by contacting appropriate student groups. We will track their progression through the course using surveys and other appropriate data collection methods and use this information to refine our methodology.

Art and Music I am also interested in generative and live coded art and music as a personal hobby. I have attended a number of live coding performances, have friends in this community, and occasionally make art and music myself. Within the academic community, I have served as a member of the program committee of the LIVE workshop and regularly attend the FARM (Functional Art and Music) workshop. I also regularly attend Moogfest, a non-academic conference and music festival for people interested in new developments in electronic music production. My research interests align well with this application domain. In particular, Hazel is a live programming environment that offers a programming model compatible with that of the popular TidalCycles live music production library, with which I have some familiarity. Uniquely, we are currently working on adding support for *palettes* to Hazel, as described in my research statement. Palettes will allow us to granularly combine conventional program editing with graphical UI elements familiar to people who have worked in art and music production: sliders, knobs, tuning curves, keyboards, dataflow diagrams, and so on. Our hypothesis is that this will help with the transition to programming for the increasingly diverse group of people interested in generative art and music production. My post-doc advisor, Ravi Chugh, and I are starting to plan a course that uses Hazel and Sketch-n-Sketch, a live graphics programming environment that our group is also developing, to teach programming using graphics and music applications. I am also interested in giving tutorials at various events, including at the university, in the local community, and at events like Moogfest, which has put substantial effort into increasing diversity.

Exposure to PL Concepts Palettes are useful for another reason: it is possible, and we conjecture reasonably straightforward, for students to define interesting new palettes specialized to the domains that they are most interested in, and in doing so, they are exposed to concepts of program representation and code generation. As such, the palettes mechanism is a useful platform for course projects and starter projects for undergraduates that might be interested in programming languages research. In combination with outreach efforts like those above and the public outreach efforts described in my teaching statement, I hope to use palettes to help broaden participation in programming languages research (which has a particularly severe problem with diversity today).

2 Onboarding and Retention

It is not enough to merely expose members of under-represented groups to programming and to programming languages research—we must also put care into subsequent onboarding and retention efforts.

PLMW and Travel Considerations I am a strong supporter of the Programming Languages Mentoring Workshop (PLMW), which is a one-day workshop co-located with every major ACM programming languages conference (POPL, ICFP, PLDI and SPLASH) that offers undergraduates and new graduate students general research advice (e.g. on how to write a good research paper) and high-level research overviews by both junior and senior faculty. As faculty, I would be interested in participating in PLMW both as a speaker and as an organizer. I am also interested in finding ways to better advertise PLMW to members of under-represented groups. Too often, diversity and inclusion work like this falls on members of under-represented groups themselves, reducing the time available for their research. I believe it is important for all of us to contribute toward these efforts.

Of course, events like these are only useful if the attendees are actually able to travel to the conference location. PLMW has been successful in making travel funding available to students for whom other sources of funding are not available, and I hope to help continue this practice. Another problem, which has become particularly acute in the last few years, has to do with visa approvals. Recently, a student from China was denied a visa to attend SPLASH because the US government decided, with no reasonable basis, that their research interest in JIT compilers was a red flag. More than half of the 200 people who sought visas to attend the Black in AI workshop at NeurIPS in Canada did not receive them on time, including several organizers. I have experienced this issue myself—I had a paper accepted at ICSE 2014, which was located in India, and I applied for a visa immediately after notification. However, the visa application was neither officially approved nor rejected at any point, so I was unable to attend and I had to record my talk. I later learned that this is a common experience for other US citizens with Pakistani ancestry who need to travel to India. In my capacity as a conference organizer, I plan to work to ensure that we identify and communicate clearly about these issues to potential attendees as early as possible, work to minimize their impact, and encourage all attendees to fight these and other discriminatory and capricious immigration policies, which can significantly impact the careers of members of targeted groups.

Communication and Community People who are new to programming or to research, particularly those who have not previously had the opportunity to participate in a highly technical community, often struggle with the unfamiliar terminology and standards of discourse to which they are suddenly exposed. I strongly believe that people who are new to our community should have the time, the resources, and the social support that they need to adjust to this new culture. Consequently, I strive to communicate with **patience, sensitivity and generosity**, and I make sure that students know that they should never be afraid to ask a question.

I am particularly sensitive to the fact that unconscious bias can manifest in subtle but important ways. For example, a recent study of faculty recommendation letters found significant gender differences in language and tone, controlling for a number of academic factors [Madera et al., *Journal of Business and Psychology*, 2018]. I believe that we must explicitly emphasize positive communication norms and work to acknowledge and counteract unconscious bias, just as we explicitly emphasize qualities like precision and rigor, if we are to build a programming and research culture that does not alienate and exclude members of under-represented groups. I believe that great universities have a cultural responsibility to lead in this regard.

I am closely following the recent efforts in many academic and open source communities to develop more supportive and inclusive community norms and codes of conduct, including codes of conduct that seek to prevent harassment of every sort. Harassment is a serious problem that can limit career opportunities at all levels, particularly for women. Research suggests that the most potent way to limit harassment is to create an organizational climate where harassment is very clearly not tolerated [Johnson et al., *The National Academies Press*, 2018]. I plan to incorporate these lessons into the Hazel community as it continues to grow, and to push for stronger community norms and codes of conduct in the other communities to which I belong.

During my years as a graduate student, I knew several women who chose to leave the graduate program citing a feeling of isolation. To combat this, I plan to encourage more collaboration and socialization between research students, both within my own research group (where the fact that there is a common research platform, Hazel, may be beneficial) and within and between research areas in the department. I also try to be conscious about inviting younger students into hallway discussions and social events at conferences. Research is ultimately a social process, and I believe that we need to approach it with a pervasive spirit of inclusion.

3 Open Access

I am a strong supporter of open access to the scientific literature. Open access is a diversity issue: it disproportionately affects people who have neither institutional access nor the resources to purchase access as needed. I presented a summary of the economics and history of open access at a graduate student seminar series at CMU, and I have supported recent efforts in the programming languages community, spearheaded in part by my graduate advisor, Jonathan Aldrich, to achieve open access for all SIGPLAN conference proceedings. So far, we have achieved this goal for three of the four major SIGPLAN conferences (POPL, ICFP and SPLASH). I have been vocal about my disapproval of PLDI's decision to reject this effort, and I plan to continue to fight for open access at PLDI and at other venues. In the meantime, I make sure that all of my own papers are freely available online.

4 Accessible Programming UIs

For people with limited mobility, programming can be inaccessible for technical reasons: text editors may be too difficult to use. If programming were more accessible to people with disabilities, it would improve their quality of life by allowing them to more precisely control the various devices and applications that they might use throughout the day, and in some cases even open up career opportunities in programming and related fields.

My research statement describes my work to develop a brain-computer interface (BCI) for people with locked-in syndrome, where there is no muscle movement at all but sensation and cognition remain essentially intact. We were able to design and build an EEG-based BCI that behaved optimally in a certain information-theoretic sense. We then incorporated statistical models of the kinds of programs that the user is most likely to write to further improve performance. We were ultimately able to test a prototype of this system on a patient with locked-in syndrome.

In that research, the programs were relatively simple control sequences for wheelchairs and speech synthesizers. I am interested in continuing this line of work to enable the communication of more sophisticated programs, which would allow further autonomy for people with severely limited mobility. My ongoing research into structured editing in Hazel provides a foundation for this project in that it defines a language of sequenced actions exactly as required by the framework that we developed. This work also provides a foundation for accessible programming UIs of other designs, e.g. for people with less severe motor deficits, who might be able to select from menus but cannot write freeform text. I am inspired in part by a graduate student with quite limited mobility who worked in the office directly across from mine for some time at CMU. With appropriate technical and human assistance, he was able to make significant research advances in proof theory and logic. I believe that we can do more to include people like him in the programming community and the computer science research community.