curriculum vitæ of

Mattox Beckman

mattox@illinois.edu +1.224.935.0444

Education

Jan. 1993 – Dec. 2003 **Ph.D.** Computer Science — University of Illinois at Urbana-Champaign

Advisor: Sam Kamin; Dissertation title: *Controlling Self-Applicable Online Partial Evaluation with Strategies*

Aug. 1989 – Dec 1992 **B.S.** in Mathematics and Computer Science — University of Illinois at Urbana-Champaign

Publications

Conference and Journal Publications

1. Yuting Chen, Hyun Eun Choi, Blake Johnson, Mattox Beckman, Lucas Anderson, Matt Goodman, Chris Migotsky, Nicole Johnson; *Integrative engineering leadership initiative for teaching excellence*; in the 2018 American Society for Engineering Education annual conference.

2. Alexandra Nilles, Mattox Beckman, Chase Gladish, Amy LaViers; *Improv: Live Coding for Robot Motion Design*; Proceedings of the 5th International Conference on Movement and Computing. (2018)

3. Nianen CHen, Yue Yu, Shangping Ren, Mattox Beckman; *A Role-Based Coordination Model and its Realization*; in Informatica v32 issue 3. (2008)

4. Mattox Beckman, Sam Kamin; *Controlled self-applicable on-line partial evaluation, using strategies*. In Proceedings of the 1998 International Conference on Computer Languages. (1998)

Workshop Publications and Posters

5. Yuting Chen, Hyun Eun Choi, Blake Johnson, Mattox Beckman, Lucas Anderson; Poster: *Integrated engineering leadership initiative for teaching excellence (IELITE) Year Two: Assessment of intermediate-term outcome for graduate teaching assistant training*; in the 2019 American Society for Engineering Education annual conference.

6. Shangping Ren, Mattox Beckman, Tzilla Elrad; *System imposed and application compliant adaptations*; in the 24th International Conference on Distributed Computing Systems Workshops, 2004.

Submitted and Pending Publications

7. Yuting Chen, Hyun Eun Choi, Blake Johnson, Mattox Beckman, Lucas Anderson; Inform Track: Integrated Teaching and Leadership Development Program for Graduate Teaching Assistants; in the 2021 American Society for Engineering Education annual conference.

8. Siwei Shen, Mattox Beckman; Students' Perceptions of Two Active Learning Strategies in a Higher-level CS Course: TPS vs POGIL; in the 26th Annual Conference on Innovation and Technology in Computer Science Education

Honors

2012–2015 Teaching Fellow — IIT

The teaching fellows program selected one teaching faculty from each department at IIT's College of Science. The committee met monthly to discuss issues related to teaching faculty and propose innovations to improve teaching at IIT. Some of the outcomes included a uniform model of dealing with cheating and plagiarism, classroom observations for new faculty (both teaching and tenure system), and helping to develop standard procedures for reviewing teaching faculty.

2006 CS Department Teacher of the Year award IIT

Grants

Fall 2018–Spring 2019 Process Oriented Guided Inquiry Learning POGIL

PI Mattox Beckman

CO-PIs Jenny Amos, Eric Shaffer, Mariana Silva

Funding Source AE3/SIIP

Process Oriented Guided Inquiry Learning (POGIL) is a team-based educational strategy that simultaneously develops process skills (e.g., team-work, critical thinking, etc.) and guides students through a process that enables them to construct their own knowledge of the topic. It has been researched extensively for over 20 years.
This team hosted two presenters from the NSF supported POGIL foundation, who held a one day workshop in December 2018 for 17 attendees. We continue to develop POGIL activities for our courses and build technology to assist facilitation.
As part of this grant we invited Brandon Myers from the University of Iowa to give a talk about POGIL in his computer science department. (November 16, 2018)

2017 – Present Integrative Engineering Leadership Initiative for Teaching Excellence iELITE

CO-PIs Mattox Beckman, Yuting Chen, Blake Johnson

Funding Source AE3/SIIP

Our team has developed a TA training course that teaches Graduate TAs skills similar to the Collins Scholars program given to new faculty in the Engineering College. We also emphasize how teaching skills transfer into industry settings.
We currently offer this training to Computer Science, Mechanical Engineering, E.C.E., and Civil Engineering. We are in discussion with other departments to offer it to their students too, with the goal of this training becoming the standard mode of training new graduate TAs.
We also have published research on the effects of this course in supporting the GTAs self-efficacy in teaching, and are exploring other topics for future work.

Service Activities

Fall 2019 – Present Faculty Senator

Fall 2017 – Present Department Github for CS Courses Pilot Github

Part of a working group to explore the use of github in the computer science department, document best practices, and encourage faculty to use it in their courses.

Fall 2016 – Present Faculty Coach International Collegiate Programming Contest (ICPC)

The International Collegiate Programming Contest (ICPC) is a premier programming competition sponsored in part by the ACM. The role of the coach is to oversee the logistics of training the student teams, holding try-outs, and transportation to the regional contests and world finals.

Fall 2019 – Present Regional Contest Site Director ICPC

Set up UIUC as a regional contest site. In 2019, 15 teams competed here, including including five from UIUC, four from Northwestern, four from U of Chicago, and one each from IIT and the College of Lake County.

Fall 2018 – Present Course Development ICPC

Created a course to teach students how to succeed in competitive programming. Students take this course to become better coders, become more confident during interviews, and to begin their careers as competitive programmers.

Fall 2019 Academic Awards Committee

Spring 2017, 2018,2020 Teaching Faulty Recruiting Committee

Spring 2020 Financial Aid and Admissions Committee

External Service Activities

March 12, 2021 **Table Mentor** SIGCSE TS 2021
Will serve as a table mentor for the Professional Development for Teaching-Track Faculty pre-symposium event

2020 **Paper Reviewer** SIGCSE TS 2021

2020 **Paper Reviewer** SIGCSE 2020

2019 **Paper Reviewer** ASEE 2019

Service Activities at Illinois Institute of Technology

Spring 2011 **IPro Review Committee** IIT
IIT has a senior capstone course called *IPro* for Inter-professional Project. Interested faculty propose projects with industry partners where a team of students chosen from diverse majors will work on a project together to solve a need for the industry partner.
This committee met regularly to review proposals from faculty, accept or reject them, and give feedback.

2009 – Aug 2015 **CS Undergraduate Studies Committee** IIT
Occasional meetings to discuss curriculum changes, ABET accreditation, and new courses.

Fall 2011–Spring 2013 **University Undergraduate Studies Committee** IIT
Campus-level undergraduate studies committee member.

2009–2015 **ACM Student Chapter faculty adviser** IIT
Campus-level undergraduate studies committee member.

2004–2015 **Co-coordinator for written Programming Language Ph.D. Qualifying Exam** IIT
Campus-level undergraduate studies committee member.

Teaching

Fall 2015 – Present **Programming Languages** CS 421
Average enrollment: 300 students. A senior level course on programming languages, including mathematical foundations and implementation. Currently working with Erin Chambers at St Louis University to incorporate some of my materials in her programming language course.

Fall 2015 – Fall 2019 **Data Structures** CS 225
Average enrollment: 800 students. Once a year I would be one of two faculty assigned to this course, typically in a support role such as running exams or lab section. I would also run an honors section to teach advanced data structures.

Fall 2017 – Present **Teaching and Leadership** CS 591 TL
Average enrollment: 120 students. I am one of three faculty leads for this course. It is similar to the Collin's Scholars faculty training program, and teaching teaching and leadership skills to new graduate teaching assistants.

Fall 2018 – Present **Competitive Algorithmic Programming** CS 491 CAP
Average enrollment: 20 students. I developed this course to train students to participate in the ICPC programming contests.

Students

Masters Thesis

Fall 2018 – Fall 2019 **Clone Detection** — Mariam Vardishvilli

Topic: Clone detection, or the detection of similar / identical code samples in Haskell and other languages. This technology can detect plagiarism and help in refactoring large programs.

Spring 2019 – Present **POGIL in Upper Level CS Courses** — Siwei Shen

Senior Thesis and Team Projects

Fall 2019 – Spring 2020 **Distributed Calendar Scheduling System** — Annie Qiu, Andrew Hsu, Scott Kim

The team created a doodle-poll like scheduling system that allows a calendar to be used among multiple appointments.

Spring 2018 – Fall 2019 **Automatic Memoization of Haskell Programs** — Victor Gao

Victor created a new compiler pass for the Glasgow Haskell Compiler (ghc) that allows for automatic memoization of functions. As Haskell is a lazy functional language with a rich type system, this proved to be very complicated. His work is near completion.

Fall 2017 **Robot Motion** — Chase Gladish

For her senior thesis, Chase Gladish worked with a graduate student Alexandra Nellis on a language to specify robot motion. They developed a domain specific language embedded in Haskell to specify robot motion in a composable way. This work resulted in a paper in MOCO 2018, which Alexandra presented.

2017 **Automated Classification of Haskell Programs** — Jia-Wern Lim, Matt Wooley, Kyle Herndon

The short Haskell programming questions for CS 421 tend to follow a few common forms, but a strict text match is likely to fail due to differences in spacing, variable names, and coding technique. This team generated a code canonicalizer for Haskell that was able to classify the students' code submissions. This allowed a grader to grade the entire classification set by examining one sample. In practice, this tended to reduce grading effort by about 40%, and we expect further improvement as we can use previous semester's data to seed the grading in the future.

2016 **Automatic Parallelization of Clojure Functions** — David Zmick

David Zmick created a tool to automate the parallelization of Clojure code. By creating a macro he was able to replace Clojure's standard `defn` function declaration keyword with his own keywords called `pardefn` and `parlet`. This macro would rewrite the function body at compile time and parallelize it. On a multi-core computer a speedup of 2–3 times was common. Hand modifying code would certainly give better results, but this allowed Clojure programmers to take advantage of multiple cores immediately simply by replacing a keyword.
This thesis was posted to the Clojure subreddit to make it available to the Clojure community.

Related External Activities

2019 – Present **Consulting** — Runtime Verification

Consulting and Haskell programming for Runtime Verification, a company that develops the K framework for proving program correctness using Matching Logic.