KEVIN A. ANGSTADT

RESEARCH STATEMENT

Research is very personal for me: I see it as a tool for answering questions I have about the world and solving problems I observe in the world around me. My research interests fall in the intersection of computer architecture, software engineering, and programming languages. Put simply, I focus on:

- Developing programming models and identifying suitable abstractions for new technologies,
- Creating tools and algorithms that ease the burden on developers for common activities, such as fault localization, debugging, and porting code, and
- Designing hardware and software architectures to address present-day challenges.

The confluence of several factors, including consumer demand, the rapid growth of data collection, and the impact of manufacturing challenges is actively reshaping our society's computing landscape. There is an increased use of hardware accelerators (e.g., FPGAs, GPUs, Google's TPU, Micron's AP, etc.) for general-purpose computing, both in server settings as well as embedded systems. Indeed, it has been reported that recent generations of Apple iPhones contain tens of different types of processors. Further, the landscape of embedded systems—ranging from smart devices and the internet of things to flight controllers on consumer-grade drones—have been exploding in numbers, kinds, and consumption in ways many might not have predicted or imagined even five years ago.

These new and emerging technologies often rely on abstractions or assumptions not supported by present techniques for programming, debugging, and securing software and systems. My research tackles this challenge head-on while involving and engaging students in the process.

RESEARCH CONTRIBUTIONS

My primary area of expertise is in the practical application of deterministic- and non-deterministic-finite state machines (DFAs and NFAs) and deterministic push-down automata (DPDA) as a hardware abstraction layer for accelerators [1–6]. These models are the same as those we teach in an undergraduate theory of computation course; my research helps students see the theory spring from the pages of proofs into the real and practical. I also have experience with the development of trustworthy hardware and software for autonomous vehicles. In this statement, I highlight my contributions that involve undergraduate research.

LANGUAGES AND TOOLS TO SUPPORT DEVELOPERS. As part of my research, I developed the first language, RAPID, supported by both Micron's AP accelerator [1] as well as more traditional architectures [2]. The language maintains the 10-100x speedup afforded by the AP over CPUs for indicative applications while providing high-level abstractions. I have also developed hardware systems that provide the abstraction of interactive, step-through debugging for RAPID programs executing (at speed) on FPGAs and Micron's AP [3]. Typically, FPGA-based debugging is extremely low-level (operating at the circuit level), but I designed algorithms to allow for setting breakpoints in—and inspecting the variables of—high-level RAPID programs, significantly easing the burden on developers who may not have training in circuit design. The lead author on this work was one of my undergraduate mentees. The student designed, implemented, and collected data for an IRB-approved human subjects study in which we found that our interactive debugger improved fault localization accuracy in a statistically significant manner ($p < 0.05$) for students. This work was well-received by the research community, with one anonymous reviewer commenting that the "human subject study of the benefits of the approach certainly stands out from the crowd."

new architectures. I also have experience designing architectures for new automata-based processors and application domains, including computer security. I mentored a team of three undergraduate students on a project that can detect hardware side-channel attacks (such as the recent Spectre and Meltdown vulnerabilities in Intel processors) in real time. I guided my students through the development process of an anomaly intrusion detection system that monitors the sequence of memory locations accessed by a program. We designed a low-overhead, in-processor hardware unit that tracks known, trained access patterns using automata and triggers OS intervention when anomalous accesses are detected. My students developed experiments to validate the hypothesis that these access patterns identify programs and conducted a large-scale evaluation (more than 2,400 program traces and 13 billion individual memory accesses) of our prototype system using core Linux utilities as well as recent security vulnerabilities (including Spectre and Meltdown). We submitted these results for review at IEEE S&P, a selective conference in security, and we plan to resubmit to ISCA, a selective conference in architecture, with two of my three mentees as lead authors.

protecting software on autonomous vehicles. I have helped lead—and involve undergraduates in—a multi-year, multi-institutional, Air Force-funded program to develop a trusted and resilient drone software platform. Our framework combines best-in-breed approaches for securing software, detecting attacks, and automatically repairing bugs in source code, allowing a vehicle to fight through security breaches and bugs to successfully complete missions. I have integrated research prototypes into a usable end-to-end system, designed and constructed rover and drone hardware with the help of students, and led three on-site demonstrations of our technologies to program managers. I also mentored an associated undergraduate senior thesis. The student developed a preliminary threat model, studied our proposed defense strategy, and published her work (as first author) in the industrial track at DSN, a selective conference in dependable systems and security [7].

mentorship, or the role of research in academia

For me, mentorship and the training of the next generation of researchers, scholars, and engineers is the primary role of research in the academy. I also value the contribution of new knowledge through research for the betterment of society, and this often informs my choice of challenges to pursue. As a graduate student, I have had the pleasure of mentoring a total of twelve undergraduate students. I have overseen smaller-scale capstone projects/senior theses as well as multi-semester projects. Two of my mentees are first authors on peer-reviewed manuscripts [3,7], and a further five are authors on papers currently under review or are working on projects that will result in publications in the near future. One of my mentees is now a PhD student at Imperial College London. Others have gone on to serve in the Peace Corps and work at startups and large firms.

Despite my success with publishing research conducted by undergraduates, I do not view manuscripts as the ultimate goal of undergraduate research. Instead, I prioritize exposing students to new opportunities, developing strong communication skills, and learning the scientific process (from identifying projects to experimental design and evaluation). Interest and motivation are more important to me than extant skills. In my experience, motivated students will allocate the time to learn—with my guidance—the skills necessary to complete their projects; however, unmotivated or overcommitted students will likely only make minimal progress. Arguably one of my most successful mentees exemplifies this: Matthew Casias began working on research with me after completing CS 1. Matt was exceptionally motivated to learn, and would meet frequently with me to develop skills necessary to be a successful researcher. The topics we covered were diverse: theoretical models of computation, hardware accelerator architectures, web development, basic statistical analysis, and technical writing (to name a few). After publishing a paper as the first author at ASPLOS (a selective conference in my field), Matt wrote to me:

> This is probably my most proud achievement in college so far. I can't thank you and Professor Weimer enough for all of your guidance and help.

After considering graduate school, Matt ultimately decided on a career in industry. I am proud of his growth and pleased that he found the best fit for his interests. Students like Matt motivate me to maintain an active research program and continue devoting time and effort to mentoring.

I also enjoy—and seek—mentorship opportunities for students with diverse identities and academic backgrounds. I have mentored a student studying kinesiology on a project using small sensors attached to shoes, and I am currently mentoring a student majoring in Computer Science and Performance Art, who is studying how students' identities impact their performance in an intermediate-level data structures course. Indeed, I enjoy mentoring projects that are of interest to the individual students because it helps me to support my students' goals and also allows me to explore a breadth of topics. For students who might not be sure where their interests lie, my personal research interests have historically admitted approachable projects [3,7]. I also attempt to scale the scope of a project to best suit an individual student's needs (some students benefit from freedom and open-endedness while others flourish with smaller tasks and more emphasis on process). I have developed close collaborators at multiple universities (e.g., CMU, Arizona State, Michigan, Virginia, etc.) as well as in industry (e.g., BBN Raytheon, Micron, GrammaTech, Clinc, etc.), which will allow me to provide additional resources to my students, such as internship connections, REU opportunities, and advice on graduate school.

FUTURE WORK

Going forward, I am excited to continue conducting research with students to improve programming support for new and emerging technologies. Guided by my research experience detailed in the previous sections, I have begun studying the porting of existing code to hardware accelerators using automata. My approach combines insights from learning theory, automata processing, and software verification, and I am interested in extending my initial work to support a broader class of programs.

Further, I am interested in the study of functional programming models and their application to developing software for accelerators. There is already a significant research effort underway to develop functional languages for hardware design (e.g., Chisel), which shows great promise. Indeed, when teaching functional languages to students, I often use analogies about circuits to help students grasp the paradigm. My particular research interests are student-focused: how can we best prepare students to develop hardware using functional paradigms, and (more broadly) how can we prepare them to be successful in a world that relies on more than just CPUs?

I am also fascinated by the relationships between student identity and performance in Computer Science courses. An unfortunate reality of our discipline is that there is an overwhelming skew toward white and Asian males. As a first step in broadening participation in computing, I plan to employ a data-driven approach to better understand how students with differing identities and educational backgrounds are affected by psychological phenomena (e.g., stereotype threat, impostor syndrome, etc.) and how these phenomena relate to student performance and decisions to pursue a degree. Further, I am interested in applying such findings to the development of teaching practices that better supports students in their pursuit of knowledge. My past experience with the IRB-approval process and initial studies of course data position me well to pursue more detailed inquiry.

REFERENCES

[1] K. Angstadt, W. Weimer, and K. Skadron, "RAPID programming of pattern-recognition processors," in *Architectural Support for Programming Languages and Operating Systems*, ASPLOS '16, pp. 593–605, 2016.

[2] K. Angstadt, J. Wadden, W. Weimer, and K. Skadron, "Portable programming with RAPID," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, pp. 939–952, April 2019.

[3] M. Casias, K. Angstadt, T. Tracy II, K. Skadron, and W. Weimer, "Debugging support for pattern-matching languages and accelerators," in *Architectural Support for Programming Languages and Operating Systems*, ASPLOS '19, ACM, 2019.

[4] K. Angstadt, J. Wadden, V. Dang, T. Xie, D. Kramp, W. Weimer, M. Stan, and K. Skadron, "MNCaRT: An open-source, multi-architecture automata-processing research and execution ecosystem," *IEEE Computer Architecture Letters*, vol. 17, pp. 84–87, Jan 2018.

[5] K. Angstadt, A. Subramaniyan, E. Sadredini, R. Rahimi, K. Skadron, W. Weimer, and R. Das, "Cache automaton," in *Proceedings of the 51st Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-51, IEEE, 2018.

[6] J. Wadden, K. Angstadt, and K. Skadron, "Characterizing and mitigating output reporting bottlenecks in spatial automata processing architectures," in *2018 IEEE International Symposium on High Performance Computer Architecture*, HPCA '18, pp. 749–761, IEEE, 2018.

[7] K. Highnam, K. Angstadt, K. Leach, W. Weimer, A. Paulos, and P. Hurley, "An uncrewed aerial vehicle attack scenario and trustworthy repair architecture," in *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop*, DSN-W '16, pp. 222–225, June 2016.