Statement of Purpose

With the world increasingly reliant on evermore complex systems of software, the challenges of software engineering have become more pervasive and impactful. Based on my experience as a research assistant to Prof. Arjun Guha and as both a paid and open source software engineer, I believe that the best way to address these challenges is by leveraging type systems, static analyses, and other such techniques within programming languages to make the lives of software engineers better. To that end, I wish to pursue a Ph.D in Computer Science and design, create, and formalize systems to protect programmers from themselves.

After starting University, I began working with Prof. Arjun Guha and was immediately thrown into the realm of the unfamiliar. My work began with a static analysis for a declarative system configuration language called Puppet. We first approached the problem trying to leverage predicate transformers, but found a more tractable approach using the Z3 theorem prover. In the summer, I continued my work by extending the static analysis tool with a program synthesis component. We continued working on the project into the Fall, and in the Spring, our paper, *Rehearsal: A Static Verification Tool for Puppet*[3], was accepted to the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI). It included our static analysis with a ready-to-use artifact in Scala as well as proofs of both soundness and completeness. At the conference, we were awarded the Distinguished Artifact Award for the tool's ease of use and extensibility.

Then, this past semester, I was in a graduate seminar on Programming Languages. As part of the seminar, we selected, read, and presented papers each week, and built a prototype of a paper. Through the readings and presentations, I explored a number of areas in programming languages including dependent types, symbolic evaluation, probabilistic programming, and more. The variety in readings and presentation styles presented me with ample opportunity to learn essential research skills, both hard and soft. As my project, I built a functional language with a polymorphic effect system based on Daan Leijen's work with row polymorphism[2]. This gave me the opportunity to study row polymorphism and learn about alternative solutions to object-oriented programming. I was also able to see how a clever and atypical type system can provide a basis for richer and more error-free programming. Overall, the seminar has reinforced my interests in type systems and static analyses by allowing me to study them in greater depth. I hope that continuing this study into graduate school will further reinforce these interests, and help me to build systems that will positively impact the state of software engineering.

After pursuing my Ph.D, I hope to stay in academia as a professor to follow my research interests while having the opportunity to mentor the next generation of great researchers. My interest in Northeastern University stems from its faculty's broad work in formal methods and programming languages. I am most interested in Prof. Amal Ahmed's work on verified compositional compilers in a multi-language world because of its focus on verifying realistic compilers[1]. In addition, I'm interested in the works of Prof. Jan Vitek and Prof. Matthias Felleisen due to their variety of interests and consistent focus on practically solving real problems[4]. Overall, Northeastern appeals to me because of its breadth of work in programming languages and its focus on practical applications. I believe that the skills and interests I have developed as an undergraduate researcher and a software engineer make me a good fit for such a community.

References

[1] Amal Ahmed. Compositional Compiler Verification for a Multi-Language World. In Delia Kesner and Brigitte Pientka, editors, *1st International Conference on Formal Structures for Computation and Deduction (FSCD 2016)*, volume 52 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:1, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[2] Daan Leijen. Koka: Programming with row polymorphic effect types. In Paul Levy and Neel Krishnaswami, editors, Proceedings 5th Workshop on *Mathematically Structured Functional Programming,* Grenoble, France, 12 April 2014, volume 153 of *Electronic Proceedings in Theoretical Computer Science*, pages 100–126. Open Publishing Association, 2014.

[3] Rian Shambaugh, Aaron Weiss, and Arjun Guha. Rehearsal: A configuration verification tool for puppet. In *Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '16, pages 416–430, New York, NY, USA, 2016. ACM.

[4] Asumu Takikawa, Daniel Feltey, Ben Greenman, Max S. New, Jan Vitek, and Matthias Felleisen. Is sound gradual typing dead? In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '16, pages 456–468, New York, NY, USA, 2016. ACM.