

Statement of Purpose

Rohan Yadav

The most unique and exciting characteristics of computer science are how results in theory can be applied and verified on modern systems, and how advances in hardware and systems open up brand new areas of theory to explore. I am most intrigued by the area where theory and practice meet, specifically parallel computing, which exists at the intersection of my interests within the broader field of computer science – theory, systems and programming languages. Developing parallel algorithms requires understanding algorithms and theory at a deeper level, to find what within an algorithm can be exploited for parallelism. Systems knowledge is paramount to translate theoretically optimal parallel algorithms into programs that are efficient on modern hardware, where constant factors cannot be ignored. To allow programmers the ability to write and reason about efficient parallel programs, innovation in programming language design and implementation is necessitated. My current research incorporates my experience in these fields to develop efficient parallel algorithms for problems, and efficient frameworks for execution of parallel languages.

I began doing research during the second year of my undergraduate studies with Prof. Umut Acar, after taking his class on parallel algorithms. I started to work on a new implementation of the `SEQUENCE` library, an abstraction for parallel operations on sets of objects, to scale better in a purely functional setting. We designed a data structure based on chunked trees to implement the library, and were able to see improved performance and scalability of programs that used the new implementation. My library was used in benchmarking sets for projects done by Umut’s research group, and I had gained my first taste of working on a problem where it was unclear if there was an answer.

Parallel algorithms continued to intrigue me, so Umut and I began to look into the open problem of whether subgraph isomorphism can be efficiently computed in parallel. Subgraph isomorphism requires checking whether a graph G_1 encodes the same structure as a subgraph of another graph G_2 . Though the problem is computationally difficult (subgraph isomorphism has been shown to be NP-complete), it has a variety of applications in biochemistry, pattern recognition, computer vision, database systems and verification. A majority of existing algorithms for isomorphism are sequential, and scale poorly as graphs increase in size, due to the exponential nature of the problem. There exists prior work in parallel subgraph isomorphism, but it has fallen short of describing solutions that are work and/or space efficient. The main challenges in developing a parallel algorithm for subgraph isomorphism stem from the difficulty of exploring an irregular search space in parallel, while maintaining both work and space efficiency. I presented a new parallel algorithm for the problem, as well as proofs of the work and space efficiency of the algorithm. Additionally, I performed an empirical evaluation of the algorithm on synthetic and real world data sets, and found that it performs well in practice, experiencing good speedup to 70 cores. We are in the process of submitting this work to SPAA’19.

To gain further experience in computer science research, I joined one of Umut’s existing projects with PhD student Sam Westrick on memory management for fork-join parallel languages. The focus of the project is to take advantage of the invariants provided by fork-join parallel programs, and structure the memory layout to reflect the structure of the program in order to perform highly efficient memory management. Existing work has implemented these ideas, and shown that they work well in practice, but is not able to bound the time and space usage of the memory management system, as it can result in unbounded time spent in garbage collection and unbounded space overhead. As part of an in-progress undergraduate senior

thesis, I am developing a garbage collection algorithm to address these challenges, and am working towards a proof of both bounded time and maximum space used by this algorithm.

In the future, I plan to dive deeper into the design and analysis of parallel algorithms, and to create frameworks for parallel programming that allow users to easily write efficient parallel programs. I want to understand existing algorithms at a deeper level, and develop parallel solutions to problems that we currently believe are hard to parallelize. I am interested in creating algorithms that are both theoretically optimal while performing well in practice. My experience with parallel computing thus far has only affirmed to me the complexity of writing efficient parallel programs, and the difficulty of converting parallel algorithms into well performing implementations. These difficulties are challenging to overcome even with experience in the field, making the benefits of parallelism largely out of reach to the average programmer. Therefore, my other goals in parallel computing research are to create systems and languages that abstract away the implementation complexities of parallelism, allowing programmers to focus on algorithm design.

In addition to research, I deeply enjoy teaching. I have been a teaching assistant every semester after my freshman year, where I have taught for CMU's introductory functional programming course, and served multiple semesters as the head teaching assistant for CMU's parallel algorithms and data structures course. Teaching other undergraduates helped me gain confidence in myself, and learn how to share my ideas with others. Additionally, managing a course staff of other undergraduates greatly improved my leadership skills, and understanding of how a course is run behind the scenes. After I felt comfortable teaching on a personal basis, I became interested in how computer science can impact the way we teach, and how we can build tools to impact more people than just one on one interactions. I have been contributing a new education platform created at Carnegie Mellon University called Diderot, which is used by almost 500 students on a daily basis. The goal of the platform is to provide a unified system for course content and peer discussions of material, as well as course management for instructors. My work on the project allows me to directly impact the education of hundreds of students, on a scale that would be impossible as a teaching assistant.

My experience with research as an undergraduate made me realize that my happiness comes from the freedom to choose problems that interest me, and the support to dedicate my time towards solving these problems. Pursuing a PhD in computer science will allow me to continue my current research in parallel computing or pursue related research opportunities in systems or programming languages as well. In the future, I want to combine my interest in research and learning with my love for teaching to become a professor, for which pursuing a PhD will help me greatly. The prospect of becoming a world expert in a topic is very exciting, and I want the opportunity to push myself to my limits to achieve such a feat.

The PhD program at Stanford caters well exceptionally well to my current research interests. I'm particularly interested in the work of Prof. Kayvon Fatahalian and Prof. Alex Aiken. Prof. Fatahalian's work in high performance and large scale computing systems align well with my interests in systems and are exciting related directions to pursue research in. Additionally, Prof. Aiken's work regarding compilers and programming systems for high performance parallel languages is appealing to my background in programming languages. The diversity and quality of research at Stanford is extremely appealing to me, and I am extremely excited to continue my career in academia at Stanford.