Eli Pregerson

epregerson@hmc.edu | github.com/Eli-Pregerson | https://dblp.org/pid/351/1082.html

Education

Harvey Mudd College - Claremont, CA **Expected May 2024**
Bachelor of Science, Computer Science and Mathematics

- National Merit Scholarship Recipient
- GPA 3.730
- Dean's List (Spring '21 - Spring '23)

Selected Coursework:

- Math: Graph Theory, Combinatorics, Single and Multivariable Calculus, Linear Algebra, Differential Equations, Discrete Mathematics, Combinatorics, Probability and Statistics, Real Analysis, Abstract Algebra, Representation Theory
- Computer Science: Algorithms(**Fall '23 Tutor**), Programming Languages, Machine Learning, Applied Logic/Automated Reasoning, Computability and Logic (**Fall '22 Tutor**), Natural Language Processing, Data Structures and Program Development, Principles of Computer Science (**Fall '21 Tutor**), Intro to CS (**Spring '21 Tutor**)

Research

CRA Outstanding Undergraduate Research award Finalist

ALPAQA, *Quantifying the Difficulty of Automated Testing* **May - Aug 2021**
Mentor: Professor Lucas Bang at Harvey Mudd College
Description: The purpose of this project was to determine the viability of a bug prediction neural network that used novel metrics developed by Prof. Bang and members of the VLab at UCSB. Our goal was to use an existing dataset of github commit messages as a proxy for the existence of bugs, and then train a neural net to predict bugs using APC and the VLab code complexity metric PReach. While working on this project I modified the outputs of the APC tool so that they would be more conducive to machine learning, removed unnecessary assumptions within the APC tool so that it could run on a much broader set of java files, and analyzed the data. At the end of this process we determined that with the dataset we had, we could not conclude anything about APC being able to predict bugs.

ALPAQA, *Recursive APC* **June 2021 - May 2023**
Mentor: Professor Lucas Bang at Harvey Mudd College
Description: The purpose of this project was to design and implement an algorithm for accurate computation of APC for recursive functions. Previously APC had been calculated using the math of regular languages and control flow graphs, but recursive calls require the math of context free grammars to properly match function call to function return. Once the code flow is abstracted as a context free grammar, the Chomsky–Schützenberger enumeration theorem can be used to get a generating function for the number of paths of different lengths. Going from the generating function to a closed form solution is a complex process that is actively being optimized.
Resulting Publications and Talk:
E. Pregerson, "Path Complexity of Recursive Functions," 2023 IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), Melbourne, Australia, 2023, pp. 308-310, doi: 10.1109/ICSE-Companion58688.2023.00083.
E. Pregerson, S. Cullen-Baratloo, D. Chen, D. Lam, M. Szostak and L. Bang, "Formalizing Path Explosion for Recursive Functions via Asymptotic Path Complexity," 2023 IEEE/ACM 11th International Conference on

Formal Methods in Software Engineering (FormaliSE), Melbourne, Australia, 2023, pp. 76-85, doi: 10.1109/FormaliSE58978.2023.00016.
S. Dissem, E. Pregerson, A. Bhargava, J. Cordova and L. Bang, "Path Complexity Correlates with Source Code Comprehension Effort Indicators," 2023 IEEE/ACM 31st International Conference on Program Comprehension (ICPC), Melbourne, Australia, 2023, pp. 266-274, doi: 10.1109/ICPC58990.2023.00041.
Plateau 2021 - Path Complexity Predicts Code Comprehension

Orrison Lab, *Computational Representation Theory* **Jan 2023 - May 2023**
Mentor: Professor Michael Orrison at Harvey Mudd College
Description: We designed algorithms for the computation of subranking and subset summary statistics in a way that maximized sharing of intermediary values and minimized the number of addition operations. We then formalized those algorithms in the language of group frames.

PRiME, *Trace Ideals over semigroup rings* **Jun 2023 - Present**
Mentor: Professor Haydee Lindo at Pomona College and Harvey Mudd College
Description: The goal of this project is to categorize the structure of trace ideals over numerical semigroup rings. Trace ideals are powerful algebraic tools that can be used to classify rings and modules. Numerical semigroups rings are a pervasive type of ring that pop up in many areas of mathematics like algebraic geometry and number theory and they have a number of properties that make them a convenient object for study with trace ideals.
Resulting Talk: MAA MathFest 2023 Research in Motion Poster Session - Trace Ideals over numerical semigroup rings.

ALPAQA, *Search synthesis anamorphisms* **Aug 2023 - Present**
Mentor: Professor Lucas Bang at Harvey Mudd College
Description: The goal of this project is to formalize an existing search synthesis algorithm in terms of anamorphism and catamorphism. The algorithm builds a decision tree based on information gain after training on a dataset, which can be formalized as an anamorphism. Then for a particular element the algorithm takes the decision tree and collapses it down into a prediction, which can be formalized as a catamorphism.

ALPAQA, *Asymptotic Path Complexity Journal Publication* **Aug 2023 - Present**
Mentor: Professor Lucas Bang at Harvey Mudd College
Description: One all encompassing publication of APC results will make the tool more accessible and running larger experiments will demonstrate its versatility. My contributions to this project are making the tool as simple and intuitive as possible and making the replication of experiments take as few commands as possible.

Additional Projects

MuddSub, *Harvey Mudd Robosub* **Sep 2020 - Jan 2022**
Description: MuddSub is a student run club that builds and programs robots for the annual Robosub competition. While on the team, I worked on the controls, reinforcement learning, simulation, and power distribution teams, and I became the lead of the controls team. Most of the work I did was for the controls team, where I was responsible for taking in instructions from the Navigation team on where the robot should go, and then figuring out how much power should be sent to each thruster.

Factor, *AARCH64 Compiler* **Sep 2023 - Present**
Mentor: Professor Christopher Stone
Description: The goal of this project is to produce an AARCH64 compiler backend for the concatenative stack based programming language Factor. Compiling factor into native AARCH64 machine instructions would

allow for factor to be fun faster on apple silicon and would make it possible to run on arm chip machines that do not have the ability to simulate x86 instructions.