Akshay Gopalakrishnan

Curriculum Vitae

"Silence is the best teacher.."

Current- PhD Computer Science

Relaxed memory accesses are used to gain substantial improvement in the performance of concurrent programs. A relaxed consistency model specifically describes the semantics of such memory accesses for a particular programming language. Such models for high level languages as well as hardware have been prey to informal specifications, conflict with common program transformations essential for the performance, render compiler mappings incorrect and bringing in additional complexities like the infamous out-of-thin-air problem. To add to this, semantics of such models are quite un-intuitive and generally have a large learning curve for new-comers. Our current focus is on identifying whether it is possible to formalize such models using program transformations. Having such a description in our eyes will make it intuitive enough for a larger audience to understand these models and make use of them wisely. In addition, we also want to investigate the advantage of having such models w.r.t compiler correctness, compositional correctness, robustness, etc.

Previous Education

2018–2021 **Masters of Computer Science - Thesis**, *McGill University*, Montreal.

Courses Taken

1. Machine Learning (COMP 652) - Fall 2018
2. Teaching in Computer Science (COMP 598) - Fall 2018
3. Compiler Design (COMP 520) - Winter 2019
4. Epistemic & Sociocultural Understanding of Computer Science (COMP 762) - Winter 2019
5. Semantics of Programming Languages - Seminar course at University of Montreal (IFT 6172) - Fall 2019
6. Meta-Programming (COMP 596) - Winter 2020

2014–2018 **Bachelor of Information Technology**, *College of Engineering*, Pune.

Relevant Courses Taken

1. Data Structures and Algorithms - 2015
2. Digital Systems - 2015
3. Discrete Structures and Graph Theory - 2015
4. Microprocessor Techniques - 2016
5. Principles of Programming Languages - 2016
6. Theory of Computer Science -2016
7. Operating Systems - 2016
8. Algorithms and Complexity - 2017
9. Assemblers and Compilers - 2017

Masters Thesis

Title *Analysis of the ECMAscript memory model: a program transformation perspective*

Supervisors Professor Clark Verbrugge

Abstract Concurrent programs have been shown to give us tremendous performance benefits compared to their sequential counterparts. With the addition of several hardware features such as read/write buffers, speculation, etc., more efficient forms of concurrent memory accesses are introduced. Known as relaxed memory accesses, they are used to gain substantial improvement in the performance of concurrent programs. A relaxed memory consistency model specifically describes the semantics of such accesses for a particular programming language. Historically, such semantics are often ill-defined or misunderstood, and have been shown to conflict with common program transformations essential for the performance of programs. In this thesis, we give a formal declarative (axiomatic) style description of the ECMAScript relaxed memory consistency model. We analyze the impact of this model on two common program transformations, viz. instruction reordering and elimination. We give a conservative proof under which such an optimization is allowed for relaxed memory accesses. We use this result to reason about the validity of reordering accesses outside loops under the same model. We conclude this thesis by eliciting the limitations of our approach, critique on the semantics of the model, possible future work using our results, and pending foundational questions that we discovered while working on this thesis.

Publications

2021 **Analysis of the ECMAScript Memory Model: A program transformation perspective**, *McGill University*, Montreal, Thesis.
Link

2022 **Reordering Under the ECMAScript Memory Consistency Model**, *Languages and Compilers for Parallel Computing (LCPC) 2020*, New York, Conference.
Paper

Research Internships

July 2020 - October 2020 **Research Fellow**, *Max Planck Institute for Software Systems (MPI-SWS)*, Virtual, Supervisor: **Dr. Viktor Vafeiadis**.
Research Work link: : Symmetry Reduction for Model Checking Relaxed Memory programs

We investigate the advantage of using symmetry reduction to improve the performance of model checking relaxed memory programs. We identified programs that adhere to certain symmetries. We use these symmetries to prove equivalence between model-checking different symmetric executions of the same program. Such equivalences help us reduce the verification of concurrent program executions.

Projects

Personal

Summer 2017 **Generic Lex (C++)**, *Holidays (solo)*, College of Engineering Pune, Bachelors.

- A basic generic lexical analyzer to define any syntax whose grammar is given by the user in the form of many regular expressions as input.
- First major exposure to actually implementing the concept of parsing.
- Project Link : GenLex

Summer 2016 **Automata Generator (C++)**, *Holidays (solo)*, College of Engineering Pune, Bachelors.

- Automata generator used to define grammars in the form of regular expressions given by the user.
- Functionalities to convert from deterministic to non-deterministic automata and vice versa and verify if given input belongs to the grammar.
- First major exposure to coding in C++, using C++ templates and recursion style programming.
- Project Link : AutomataGen

Summer 2016 **Assembler for 8086 (C)**, *Holidays (solo)*, College of Engineering Pune, Bachelors.

- A full fledged assembler for a subset of 8086
- Involved around 10 instructions of 8086

Course-based

Winter 2022 **Constraint driven Scheduling of fine-grained C concurrency for Reconfigurable Hardware**, *Course: High level Synthesis of Digital Systems -COMP 764 (Solo)*, McGill University, PhD.

- Addressed the scheduling problem of HLS designs using fine grained concurrent constructs.
- Proposed an optimization to address resource constraints while synthesizing such designs as hardware.
- Project link: HLS-Project.

Winter 2020 **Kripke Style Interpretation of lambda circle**, *Course: MetaProgramming Course project - COMP 596 (solo)*, McGill University, Masters.

- Attempted to represent the "next" temporal logic operator using Kripke Style semantics.
- Experience in experimenting with the relation between modal logic and temporal logic.
- Project report link: KripkeLambdaCircle.

Fall 2019 **Extending Typer with Linear Types**, *Course: Semantics of Programming Languages - IFT 6172 (solo)*, University of Montreal, Masters.

- Attempted to extend an existing language with Linear Types.
- Gained experience reading compiler code written in OCAML.
- Gained experience in working with Functional programming languages.
- Project report link:TyperExtLinearTypes.

Winter 2019 **Vocabulary and its Influence on Computer Science Research**, *Course: Epistemic and Sociocultural Understanding of Computer Science - COMP 766 (solo)*, McGill University, Masters.

- Wrote a research paper that talks about how vocabulary in CS research helps in influencing CS education across different communities.
- Inspired by Pierre Bourdieu's work on "Reproduction in Education, Society and Culture".
- Link to paper : Vocabulary in CS research and its Impact

Winter 2019 **Compiler for GoLite (OCAML)**, *Course: Compiler Construction - COMP 520 (project of 2)*, McGill University, Masters.

- A full fledged compiler for a subset of GO language.
- Tremendous practical experience building each phase of compiler viz Lexical, Parsing, Semantics and Code gen.
- Tremendous experience gained in using OCAML language to build the compiler.
- The target language was Python.
- Project Link with report: GoLite Compiler.

Fall 2018 **Foundations of Programming for Grade 6 students**, *Course: Teaching in Computer Science - COMP 598 (project of 4)*, McGill University, Masters.

- Designed an entire course for grade 6 students for programming foundations
- Designed and presented lectures for a subset of the designed course throughout the term.
- Designed rubrics to grade assignments and work done by peers on designing other computer science courses

Fall 2018 **Predicting Compiler Optimizations in C (bash, C, Python)**, *Course: Machine Learning - COMP 652 (solo)*, McGill University, Masters.

- Research based project wherein gcc compiler optimization levels were attempted to be predicted by training a simple machine learning model
- Data set was gathered from my own personal C programs written number theory and a few project euler problems.
- Experience gained in using autoencoders.
- Experience gained in data mining for machine learning purposes.
- Critical insights were gained on analysis of training data and predictions.
- Project Link with report : Compiler Optimization Prediction

2017-2018 **Ontology Based Intrusion Detection System (SPARQL, Python)**, *Course: Undergraduate Final Year (project of 3)*, College of Engineering Pune, Bachelors.

- Research based project involving building an proof of concept Intrusion Detection System (IDS) for a specific application layer based Denial of Service (DoS) attacks using HTTP protocol called as SlowDos.
- Experience gained in reading research papers, conducting exhaustive literature review and concretely defining research statement for our problem.
- Research work link : Ontology-based IDS.

2016 **Interactive Debugger and Interpreter for 8086 (Python)**, *Course: Principles of Programming Languages (project of 3)*, College of Engineering Pune, Bachelors.

- An interactive debugger for Assembly Language incorporated with GUI crafted using PyQT4 library.
- Involved around 20 instructions of the 8086 instruction set.
- Project Link : Visualemu 8086

2015 **Project Othello (C)**, *Course: Data Structures (solo)*, College of Engineering Pune, Bachelors.

- First major programming project as a two player board game known as Othello (or Reversi).
- Common effective strategies of the game integrated as an AI opponent.
- Ncurses graphics library used for the visual aspects.
- Project Link : Project Othello.

Teaching Experience

Vocational

2012–Present **1st Year Analyst**, Lehman Brothers, Los Angeles.

Developed spreadsheets for risk analysis on exotic derivatives on a wide array of commodities (ags, oils, precious and base metals), managed blotter and secondary trades on structured notes, liaised with Middle Office, Sales and Structuring for bookkeeping.

Detailed achievements:

- Learned how to make amazing coffee
- Finally determined the reason for PC LOAD LETTER:
  - Paper jam
  - Software issues:
    - Word not sending the correct data to printer
    - Windows trying to print in letter format
  - Coffee spilled inside printer
- Broke the office record for number of kitten pictures in cubicle

2011–2012 **Summer Intern**, Lehman Brothers, Los Angeles.

Rated "truly distinctive" for Analytical Skills and Teamwork.

Miscellaneous

2010–2011 Spent some time finding myself. This was a courageous endeavour that didn't have a job title. It was quite important to my overall development though so I'm adding it to my CV. Also it explains the gap in my otherwise stellar CV.

2009–2010 **Computer Repair Specialist**, *Buy More*, Burbank.

Worked in the Nerd Herd and helped to solve computer problems. Allowed me to become expert in all forms of martial arts and weaponry.

Awards

2011 School of Business Postgraduate Scholarship

2010 Top Achiever Award – Commerce

Computer skills

Basic	JAVA, Adobe Illustrator
Intermediate	PYTHON, HTML, LaTeX, OpenOffice, Linux, Microsoft Windows
Advanced	Computer Hardware and Support

Communication Skills

2010 Oral Presentation at the California Business Conference

2009 Poster at the Annual Business Conference in Oregon

Languages

English	**Mothertongue**	
Spanish	**Intermediate**	*Conversationally fluent*
Dutch	**Basic**	*Basic words and phrases only*

Interests

- Piano
- Cooking
- Running
- Chess
- Dancing