Pritam Gharat
Microsoft Research, India

Email t-pgharat@microsoft.com
pritam01gharat@gmail.com
Web https://pritammg.github.io/

Research Interests

Programming languages, Software Engineering, Compilers, Program Analysis.

Education

- **Ph.D.** in Computer Science & Engineering, IIT Bombay *(June 2013 - July 2018)*
  Thesis Title: *Generalized Points-to Graph: A New Abstraction of Memory in Presence of Pointers*
  Advisor: Prof. Uday P. Khedker
- **M.Tech.** in Computer Science & Engineering, IIT Bombay *(July 2011 - June 2013)*
- **B.E.** in Computer Science & Engineering, Mumbai University *(June 2006 - May 2010)*

Experience

- **Post-Doctoral Researcher** in Microsoft Research, India *(September 2021 - Present)*
- **Research Associate** in Department of Computing, Imperial College *(August 2018 - July 2021)*
- **Teaching Assistant** in Department of Computer Science and Engineering, IIT Bombay*(August 2011 - August 2018)*
  I have worked as a TA for courses such as Computer Programming and Utilization, Implementation of Programming Languages, Design and Implementation of Gnu Compiler Generation Framework, Program Analysis, Advanced Compilers. I also worked as a TA for the workshop on Essential Abstractions in GCC and Winter School in Software Engineering.
- **System Administrator**, Amdocs DVCI, Pune *(July 2010 - June 2011)*.

Honours and Distinctions

- TCS Research Fellowship *(July 2013 - July 2018)*.
- Sir Ratan Tata Trust Merit scholarship for two consecutive years - *2008* and *2009* for excellence in academics.
- Best Student Award from Tata Consultancy Services *(2010)*.

Publications

- "Combining Static Analysis Error Traces with Dynamic Symbolic Execution (Experience Paper)", Frank Busse, **Pritam M. Gharat**, Cristian Cadar, Alastair Donaldson. ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA) 2022.
- "Generalized Points-to Graph: A New Abstraction of Memory in Presence of Pointers", **Pritam M. Gharat**, Uday P. Khedker, Alan Mycroft. ACM Transactions on Programming Languages and Systems (TOPLAS) 2020.
- "Flow- and Context-Sensitive Points-to Analysis using Generalized Points-to Graphs", **Pritam M. Gharat**, Uday P. Khedker, Alan Mycroft. 23rd Static Analysis Symposium (SAS) 2016.
- "CoS-SSA: SSA for Context-Sensitive Interprocedural Analysis", **Pritam M. Gharat**, Uday P. Khedker, Alan Mycroft – Under preparation.

Projects

- CoS-SSA: Context-Sensitive Interprocedural SSA
  (In Collaboration with Prof. Uday Khedker and Prof. Alan Mycroft (September '21 - Present))
  - The goal of this work is to construct an interprocedural SSA form, called the *context-sensitive SSA* (aka CS-SSA), for scalars and pointers that may be global or address-taken local such that a context-insensitive and flow-insensitive analysis over the CS-SSA form of a program yields the same results as that of a context-sensitive and flow-sensitive analysis of the program.

- We achieve this by creating context-sensitive versions of variables depending on the definitions that reaches the uses of the variables along different contexts. Obtaining the version of a variable for a context requires discovering points-to information that holds along the context.

- **Combining Static Analysis Error Traces with Dynamic Symbolic Execution**
  (Postdoctoral work at Imperial College, May '19 - July 2021)
  - This project automates the process of confirming potential bugs reported by static analysis and generating concrete input to trigger the confirmed bugs.
  - The idea is to apply a *dynamic symbolic execution* (DSE) tool to the program that is instrumented to contain the static information generated by the analyser. The DSE tool explores only those paths that agree with the static information with the aim of confirming the specific bug reported by the analyser.
  - Our hypothesis is that if the bug turns out to be a true positive then the DSE tool may be able to confirm the bug (producing an associated triggering test case) more efficiently than if it were run on the program in a default, undirected fashion.

- **Generalized Points-to Graph: A New Abstraction of Memory in Presence of Pointers**
  (Ph.D. Thesis under the guidance of Prof. Uday Khedker, June '13 - July '18)
  - This work proposed the concept of Generalized Points-to Graphs (GPGs) as a new representation of procedure summaries for scalable flow- and context-sensitive points-to analysis.
  - By construction, GPGs are compact and yet bottom-up precise procedure summaries. The main challenge in this is to handle the indirect assignments to variables through pointers that are defined in the callers. This requires retaining control flow which makes the summaries too large. Ignoring control flow could make the summaries unsound. Overapproximating control flow could make the summaries imprecise. GPGs achieve the seemingly conflicting goals of compactness and precision simultaneously, by retaining just enough control flow information so that soundness is not violated and no loss of precision occurs.
  - The implementation for GPG-based flow- and context-sensitive analysis scaled to 158kLoC for C programs.

- **Improving Interprocedural Analysis**
  (M.Tech + Ph.D. Dual Degree Research Proposal, guided by Prof. Uday Khedker, Oct '12 - Dec '12)
  - Reforming Value Based Call Strings Method by eliminating the re-processing of flow functions and improving the efficiency.
  - Proposed a variant to k-CFA called as Var-k-CFA for higher order languages by building an analogy between Var-k-CFA and Value Based Call Strings Method.

- **Static Analysis of Object Oriented Languages**
  (M.Tech. Seminar, guided by Prof. Uday Khedker, Jan '12 - May '12)
  - A study of several static analyses designed for Object Oriented Languages that include Class Hierarchy Analysis, Pointer Analysis, Escape Analysis, Type Analysis.

- **Discovering use of Pointer Information in GCC** *(GCC Project, Spring 2011)*
  - Points-to information generated by Pointer Analysis is used by other static analyses to perform further optimizations for better precision. The goal of the project was to discover how the points-to information is used by other optimizations in GCC.

Service

- CGO 2021 (Artifact evaluation)
- ISSTA 2021 (Artifact evaluation)
- PLDI 2020 (External review committee)
- SAS 2020 (Artifact evaluation)