## **Objectives**

Seeking research opportunities in the broad area of programming languages. Current experience is in Compilers, Program Analysis.

### Education

- **Ph.D.** in Computer Science & Engineering, IIT Bombay (*June 2013 Present*) 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)

### **Industrial Experience**

- System Administrator, Amdocs DVCI, Pune (July 2010 June 2011).
- Worked on Technologies like HP DataProtector, HP MC ServiceGuard for Clustering.

### **Honours and Distinctions**

- 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).
- TCS Research Fellowship (July 2013 Present).

## **Publications**

- "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.
- "Generalized Points-to Graph: A New Abstraction of Memory in Presence of Pointers", **Pritam M. Gharat**, Uday P. Khedker, Alan Mycroft. Computing Research Repository (CoRR) arXiv:1801.09189 [cs.PL], 2018. (Submitted to ACM Transactions on Programming Languages and Systems (TOPLAS) 2018).

#### An Overview of the Work Done in Ph.D.

My Ph.D. thesis advocates a new form of bottom-up procedure summaries, called the generalized points-to graphs (GPGs) for flow- and context-sensitive points-to analysis.

GPGs can represent both memory transformers that summarize the effect of a procedure and the resulting memory in terms of points-to relations. GPG contains GPUs (generalized points-to updates) representing individual memory updates along with the control flow between them. GPUs uses the counts of indirection levels and hence leaves the unknown pointees accessed indirectly through pointers implicit.

GPGs are compact; their compactness is achieved by a careful choice of a suitable representation and a series of optimizations: strength reduction reduces the indirection levels, redundancy elimination removes redundant memory updates and minimizes control flow (without over-approximating data dependence between memory updates), and call inlining enhances the opportunities of these optimizations. These optimizations are similar to the optimizations performed by compilers.

These optimizations uses operations such as GPU composition and GPU reduction and data flow analyses such as reaching GPUs analysis and coalescing analysis. Essentially, these optimizations eliminate the data dependence between memory updates by rewriting the memory updates thereby discarding redundant control flow whenever possible.

Our quest for scalability of points-to analysis leads to the following insight: The real killer of scalability in an analysis is not the amount of data but the amount of control flow that it may be subjected to in search of precision.

The effectiveness of GPGs lies in the fact that they discard as much control flow as possible without losing precision (i.e., by preserving data dependence without over-approximation). The implementation for GPG-based flow- and context-sensitive analysis scaled to 158kLoC for C programs.

## **Experience as a Teaching Assistant**

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.

## **Projects**

- **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 reprocessing 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.
- Implementation of GVN PRE (Advanced Compilers, Autumn 2012)
  - Implemented GVN PRE in Jikes RVM. Studied dynamic compilation and adaptive optimization system in Jikes RVM.
- Implemetation of various classification, clustering and apriori Machine learning algorithms (Machine Learning, Autumn 2011)
  - Implemented Non-probabilistic classifier ID3 and a Probabilistic Classifier Naive Bayes, Agglomerative Hierarchical Clustering, K-means Clustering.
  - Tested these implementations on 20 News groups dataset http://people.csail.mit.edu/jrennie/20Newsgroups/ and computed their accuracy percentages.
- Message Passing Protocol (Advanced Machine Learning, Spring 2011)
  - Performed Triangulation and Junction Tree creation for Undirected Graphs and implemented Message passing algorithm for MAP and Sum Query on Graphical Models.
- GIS Network Analysis (B.E. Project, 2010)
  - Implemented algorithms for Service Area, Closest Facility, Traveling Salesman Problem, Location-Allocation Facility in Network Analysis Module of GRAM++, a GIS based software developed at Bhugol GIS Pvt. Ltd., CSRE, IIT Bombay.
  - Enhanced Network Analysis module for solving spatial problems such as Transportation, Resource Allocation and Distribution.

# Certifications

- Red Hat Enterprise Linux (RHEL) 5.5 Certified Engineer
- HP-UX 11i v3 Certified System Administrator
- IBM Certified System Administrator AIX 6.1