Saurabh Joshi

CS213,Department of CSE
IIT Kanpur
Kanpur - 208016
Uttar Pradesh, India

Phone: To be provided on request
sbjoshi@cse.iitk.ac.in

Education

PhD in Computer Science (2007 - Present)

University	:	Indian Institute Technology Kanpur
Advisors	:	Prof Sanjeev K Aggarwal
		Prof R K Shyamasundar
Cumulative Performance Index	:	9.5/10

MTech (as Research Assistant) in Computer Science (2003 - 2006)

University	:	Indian Institute of Technology Bombay
Advisor	:	Prof Supratik Chakraborty
Cumulative Performance Index	:	8.84/10

BE in Information Technology (1999 - 2003)

University	:	Sardar Patel University
Advisor	:	Prof Ketan Kotecha
Cumulative Performance Index	:	8.78/10

Patent(s)

1. "*Finding Bugs with Low False Alarms and Underspecified Harness*"
   Filed for US patent (through Microsoft Research), 2011

Publication(s)

1. "*Automatically Finding* `atomic` *Regions for Fixing Bugs in Concurrent Programs*"
   with Akash Lal
   19th International Static Analysis Symposium (SAS) 2012 (Submitted)

2. "*A New Method of MHP Analysis for Languages with Dynamic Barriers*"
   with R K Shyamasundar, Sanjeev Aggarwal
   17th International Workshop on High-Level Parallel Programming Models and Supportive Environments (HIPS), IPDPS Workshops, 2012 (To appear)

3. "*Underspecified Harness and Interleaved Bugs*"
   with Shuvendu Lahiri, Akash Lal
   39th Symposium on Principles of Programming Languages (POPL 2012)

4. "*Distributed Generalized Dynamic Barrier Synchronization*"
   with Shivali Agrawal and R K Shyamasundar
   12th International Conference on Distributed Computing and Networking (ICDCN 2011)

5. "*Reactivity in SystemC Transaction-Level Models*"
   with Frederic Doucet, R K Shyamasundar, Ingolf Krueger and Rajesh Gupta
   Haifa Verification Conference (HVC 2007)

Research Experience

1. **Verification and repair of concurrent programs at Microsoft Research India (May 2010 - Sep 2010, Feb 2011 - Apr 2011)** :
   I collaborated with Rigorous Software Engineering group at Microsoft Research India during my internships to work on two different problems. We provided a technique to precisely locate interleaved bugs for concurrent programs even when the given harness is underspecified. This work resulted in a publication at POPL 2012 as well as filing of a patent. Another work is on automatically finding a set of smallest atomic regions to repair a given concurrent program with respect to given properties. This work will be submitted soon.
2. **Improving precision and usability of static analysis tools at IBM India Research Lab (2006-2007)** :
   Different static analysis tools have different strengths and weaknesses in finding different kinds of bugs. A true bug reported by a tool often gets buried in an avalanche of false positives rendering the tool extremely difficult to use. Using many tools in the background - transparent from the user - and then selecting the meaningful results from known strengths of tool as well as past experience of the user can be helpful. During the process, internal analysis techniques of a few tools were improved.

Internships

1. Internship at Microsoft Research India, Bangalore (Feb 2011 - Apr 2011)
2. Summer Internship at Microsoft Research India, Bangalore (May 2010 - Sep 2010)
3. Two week summer school on Programming Languages, Analysis and Verification by Microsoft Research, held at the Indian Institute of Science, Bangalore during June 16 - June 28 2008.

Teaching Assistantship/Mentorship

1. Teaching assistant for **CS738 - Advanced Compiler Optimizations** (Instructor : Dr Amey Karkare), Jan 2012-Present
2. Project mentor for **CS100 - Sudoku Solver using a SAT Solver**, guided a group of BTech students for this course project, Jan 2010-April 2010
3. Project mentor for **CS100 - A Toy SAT Solver**, guided a group of BTech students for this course project, Jan 2009-April 2009
4. Teaching assistant for **CS719 - Data Streaming Algorithms** (Instructor: Prof Sumit Ganguly), July 2008-Dec 2008

Other Academic Activities

1. Reviewer: High Performance Computing (HiPC) 2010
2. Reviewer: Principle and Practices of Parallel Programming (PPoPP) 2010
3. Talk : "Interleaved Bugs and Underspecified Harness", Mysore Park Workshop on The Future of Debugging, Mysore, 2012.

References

To be provided on request.

Brief description of research works

1. "*Automatically Finding* `atomic` *Regions for Fixing Bugs in Concurrent Programs*"
   with Akash Lal
   To be submitted soon.

   Finding bugs due to atomicity violation in concurrent programs is difficult. Fixing them is even more so. In this work we statically find the smallest set of atomic regions needed to fix the program

with respect to the properties specified in the program. We provide algorithms with correctness guarantees. Using these algorithms, our tool finds the atomic regions with respect to strong as well as weak atomicity semantics. We are getting encouraging results on benchmark programs.

2. "*A New Method of MHP Analysis for Languages with Dynamic Barriers*"
   with R K Shyamasundar, Sanjeev Aggarwal
   17th International Workshop on High-Level Parallel Programming Models and Supportive Environments (HIPS), IPDPS Workshops, 2012 (To appear)

   May-happen-in-parallel analysis enables various optimizations and effective debugging of concurrent programs. For languages with dynamic barriers, where activities (or threads) can join or leave barrier synchronization dynamically, precise MHP analysis becomes a bit more difficult. We propose *Phase Interval Analysis* (PIA) which conservatively gives a bound on the phases in which a given statement may execute. This enables us to infer an order between two given statements. Ordering information between two statements provide more optimization opportunities as opposed to just inferring that they will not execute in parallel. In addition, using the transitive property of the ordering relation, we can detect indirect synchronization happening between two statements in absence of any common barrier between the two.

3. "*Underspecified Harness and Interleaved Bugs*"
   with Shuvendu Lahiri, Akash Lal
   39th Symposium on Principles of Programming Languages (POPL 2012)

   Precise verification of open programs not only depends on a precise model checker but also on the precision of the harness which models the guarantees provided by the environment. The process of writing a precise harness, involves lots of iterations and manual effort. We automate this process to find interleaved bugs for concurrent programs by using the program behaviour under sequential semantics as a classifier to rule out the bugs due to imprecise harness. We justify using the sequential behaviour as a guideline due to several practical observations mentioned in the paper. We show effectiveness of our method on real world programs.

4. "*Distributed Generalized Dynamic Barrier Synchronization*"
   with Shivali Agrawal and R K Shyamasundar
   12th International Conference on Distributed Computing and Networking (ICDCN 2011)

   Modern languages for parallel programming provide rich synchronization constructs like dynamic barriers, where processes can join or leave the barrier dynamically. This paper describes a distributed protocol to achieve barrier synchronization in a dynamic setting. It also provides progress and starvation freedom guarantees.

5. "*Reactivity in SystemC Transaction-Level Models*"
   with Frederic Doucet, R K Shyamasundar, Ingolf Krueger and Rajesh Gupta
   Haifa Verification Conference (HVC 2007)

   SystemC is a popular language used in modeling system-on-chip implementations. To support this task at a high level of abstraction, transaction-level modeling (TLM) libraries have been recently developed. While TLM libraries are useful, it is difficult to capture the reactive nature of certain transactions with the constructs currently available in the SystemC and TLM libraries. In this paper, we propose an approach to specify and verify reactive transactions in SystemC designs. Reactive transactions are different from TLM transactions in the sense that a transaction can be killed or reset. Our approach consists of: (1) a language to describe reactive transactions that can be translated to verification monitors, (2) an architectural pattern to implement reactive transactions, and (3) the verification support to verify that the design does not deadlock, allows only legal behaviors and is always responsive.

6. "Symbolic Model Checking of Large Sequential Circuits"
   MTech Thesis (Advisor : Prof Supratik Chakraborty)

Performing exact state reachability analysis over large state space is very difficult. Some of the symbolic reachability approaches fail due to blow up in representation of state-space even in a symbolic form. Reachability analysis over decomposed state space using combinational circuit as a representation was the goal of this thesis. SAT based techniques were employed to keep the experiments within the time and space limits. We had a partial success in this direction.