About Me

Name: Sandeep Dasgupta
Degree(s) received: PhD (Computer Science) from UIUC
Field of expertise: Compilers and computer Architecture verification
Current Affiliation: Google Research
Webpage: https://sdasgup3.github.io/

Journal or paper (any media outlet) ever written about me and my contributions to the field?

Google Citations: Total 45 citations

My recent works on "formalizing X86-64 semantics" and "Scalable Binary Lifters" has been referred to by 34 articles (conference papers and journals) and a blog article and PhD dissertation.. Additionally, I have 11 more citations on other publications.

Articles written by me pertaining to your field of expertise

6 Google Citations
(Note: 1. The workshop paper is peer-reviewed and 2. The two "Precise shape analysis using field sensitivity" are separate peer-reviewed papers, and the count 6 does not include my PhD published thesis)

Member of any associations which require outstanding achievement of their members

1. I am in the program committee of PLDI'20. This is a highly prestigious conference in the domain of programming languages and system design and selection criterion is based on extraordinary achievements in the area of expertise, which in my case is compilers and reverse engineering tools. Every year, a total of ~127

people from all over the world are invited, through invitation from the program chair, to serve as program committee members.

2. I am a part of the Google Research Team that tackles challenges that define the technology of today and tomorrow. There are in total 4000 employees in the group and are chosen based on established achievements in research and merit evaluated based on multiple interviews which focus intensively on the depth and breadth of knowledge in your domain of expertise.

Served as the judge of the work of others in your industry?

I have reviewed more than 30+ papers in various journals and conferences. I am associated with the following conference/journal bodies.

1. Program Committee Member at the 42nd ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2021) [PLDI 2021] (11 paper reviews)

2. Sub-reviewer, The 30th International Conference on Concurrency Theory [CONCUR 2019] (1 review)
3. Sub-reviewer, Workshop on Instruction Set Architecture Specification [SpISA 2019] (1 review)

4. Editorial Board Member of American Journal of Neural Networks and Applications

5. Technical Program Committee Member for (20+ reviews)
– The International Journal of Artificial Intelligence [IJAI 2020]
– TELKOMNIKA (Telecommunication Computing Electronics and Control) [TELCOMNIKA 2021]
– International Journal of Electrical and Computer Engineering [IJECE 2021]
– The 1st Conference on Internet of Things and Embedded Intelligence [CITEI 2020]
– Bulletin of Electrical Engineering and Informatics [BEEI 2020-21]
– International Conference on Computer Applications & Information Security [ICCAIS 2020-21]
– International Conference on Artificial Intelligence & Modern Assistive Technology [ICAIMAT 2020]
– International Journal of Informatics and Communication Technology [IJ-ICT 2020]
– International Journal of Robotics and Automation [IJRA 2020]

Work ever been displayed at artistic exhibitions or showcases?

1. Presented [VID and Presentation Flyer and PPT] my work at Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '20), June 19, 2020.
2. Presented [VID and Presentation Flyer and PPT] at Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '19), June 26, 2019.
3. Presented at Workshop on Instruction Set Architecture Specification (SpISA 2019), September 13, 2019. [Link]
4. Invited talk at Workshop on Declarative Program Analysis (DPA) June 23, 2019. [Link]
5. Presented at LLVM Developers Meet, Nov 4, 2016. [Link]
6. Presented at 8th ASIAN Symposium on Programming Languages & Systems, APLAS 2010, Nov 2010. [Link]

Awards

● Awarded University Gold Medal for securing 1st Rank in B.Tech, Computer Science & Engineering. This award is given every year to one person demonstrating stellar academic achievement.
● Awarded Best Student Award, sponsored by Tata Consultancy Services (https://www.tcs.com/), for outstanding performance in BE, Computer Science & Engineering. This award is given every year to one person demonstrating stellar academic achievement.

How my contributions are new or unique to the field of expertise

Contribution 1:
Processor details are documented in the hardware-vendor provided manuals, formally known as Instruction Set Architectures (ISA), which defines the detailed behaviour and capability of the binary-code (a set of instructions represented in 0s and 1s) which runs on a particular processor. Intel's x86-64 instruction set architecture (ISA) is one of the most complex and widely used ISAs on servers and desktops, and it defines the expected behaviour of binary code which runs on the Intel processors. It is possible that the execution behaviour of a particular instruction deviates from that defined in the manual. Such a deviation means either the processor's implementation of that instruction execution is faulty or the documentation in the manualis incorrect. In either case, it is imperative to catch those deviations and report it to the concerned authorities. One of the challenges in comparing the manual and the processor's implementation is that the documentation provided by the hardware manufacturers are informally specified and it runs into thousands of pages and has ambiguities, missing specifications, or even bugs.

Because of that there is no way to automatically compare the behaviour of an instruction based on what is defined in the manuals against what the processor implements. The only option is to manually compare the behaviour which is infeasible as there are thousands of documented instructions and each instruction can exhibit hundreds of thousands of behaviours.

One way to catch the bugs is to create an alternative “Executable & Formal” documentation of the ISA. “Executable” means that the documentation can be executed and hence the behaviour can be compared against that of a processor. It is “Formal” meaning that we can draw logical reasoning on it. We call such documents as the formal semantics of the ISA. The x86-64 instruction set architecture (ISA) is one of the most complex and widely used ISAs on servers and desktops. Given its importance, there have been many previous efforts in defining the formal semantics of Inel’s ISA but they are limited in coverage and defined just a subset of the actual ISA.

One of my contributions is to have the first successful attempt in defining the complete formal semantics of Intel’s ISA and make sure, using thorough testing, that it is correctly representing the manual and can be used as an alternative for that. This contribution is published in one of the premier conferences the world and acknowledged worldwide. The work helped us find unexpected bugs in the processor manuals which helped the community revise their documents.

Contribution 2:
The code which actually runs on processors, known as binary code or just binary, is encoded using 0 and 1s and it is not human readable. There are external tools, known as lifters, which convert (or “lifts”) this code (a list of instructions) to a human readable format. Such a lifting is useful because it allows reasoning about the otherwise illegible binary code. Such reasoning is important to discover opportunities to improve the binary, or to find if there is some bad behaviour injected into the code by some malicious attackers.

The way the lifter functions is that they convert the binary instructions, one at a time, to a human readable format. The converter needs to understand the detailed behavior of each instruction which in fact is achieved manually by the developers of a lifter by encoding the effects of all the instructions that the lifter expects to lift. This is challenging because manually encoding the effects of a vast number of instructions are hard. Because of the nature of how the instructions are manually encoded, there could be potential bugs in the encodings. It is imperative, for the correct functioning of the lifters, that the encodings are correct so that the converted high level format can be relied upon.

My next contribution is towards ensuring the correctness of the lifters. In order to accomplish that we leverage the formal semantics we developed, as part of my first contribution, to establish the correctness of the lifters. The idea is to thoroughly compare the encoding of the lifters with the formal semantics that we developed. Our approach, in ensuring the correctness of lifters, is unique and better than all the previous approaches. The claims about the

uniqueness and superiority of our approach is evaluated by one of the premier conferences in the world.

My first contribution helped Intel, a big name in processor developers, revise their ISA manuals. We reported several discrepancies in their manuals which they not only acknowledged but also fixed.

My second contribution helped the developers of McSema, a widely used lifter, to revise their instruction-encodings so that they can improve their task of converting the binary code to human readable format.

I can provide evidence for all of the above.