# DANESHVAR AMROLLAHI

☐ daneshvar@cs.stanford.edu ( cs.stanford.edu/~daneshva ( github.com/daneshvar-amrollahi 353 Jane Stanford Way, Gates Computer Science #481, Stanford, CA 94305, United States

#### **EDUCATION**

• Stanford University

2024/01 - Present

PhD in Computer Science (Advisor: Clark Barrett)

• University of Tehran

BSc in Computer Engineering (Software)

2018/09 - 2023/02

GPA: 18.02/20.00

# RESEARCH INTERESTS

• Automated Reasoning

• Satisfiability Modulo Theories (SMT)

• Verification

• Computer Systems

# **PUBLICATIONS**

- D. Amrollahi, E. Bartocci, G. Kenison, L. Kovács, M. Moosbrugger, M. Stankovič (2022). Solving Invariant Generation for Unsolvable Loops. 29th International Static Analysis Symposium (SAS 2022). Awarded the Radhia Cousot Young Researcher Best Paper Award.
- A. Humenberger, D. Amrollahi, N. Bjørner, L. Kovács (2022). **Algebra-Based Reasoning for Loop Synthesis**. Formal Aspects of Computing (FAC).
- D. Amrollahi, H. Hojjat, P. Rümmer (2023). **An Encoding for CLP Problems in SMT-LIB**. 10th Workshop on Horn Clauses for Verification and Synthesis (HCVS 2023).
- D. Amrollahi, E. Bartocci, G. Kenison, L. Kovács, M. Moosbrugger, M. Stankovič (2023). (Un)Solvable Loop Analysis. Submitted to Formal Methods in System Design (FMSD).
- P. Hozzová, D. Amrollahi, M. Hajdu, L. Kovács, A. Voronkov, E.M. Wagner (2024). Synthesis of Recursive Programs in Saturation. *International Joint Conference on Automated Reasoning (IJCAR 2024)*.

#### RESEARCH EXPERIENCE

• Center for Automated Reasoning, Stanford University Under Prof. Clark Barrett

Stanford, United States 2024/01 - Present

Working on the performance robustness of cvc5, in response to semantics-preserving query mutations (assertion shuffling, symbol renaming, operand re-ordering, etc).

• Research Intern at Automated Program Reasoning Group, TU Wien

Vienna, Austria

Under Prof. Laura Kovács and Prof. Ezio Bartocci

Worked on different topics including polynomial loop invariant generation, program synthesis, symbolic computation, probabilistic programming, saturation-based theorem proving, structural induction, etc.

• Research Intern at Dependable Systems Lab, EPFL Under Prof. George Candea

Lausanne, Switzerland 2022/07 - 2022/08

Integrated Z3's support for quantifiers in first-order logic into KLEE's source code, to mitigate the path explosion issue in symbolic execution due to loops (e.g., libc strings functions), by using loop summaries.

• Research Intern at Programming Methodology Group, ETH Zürich

Zürich, Switzerland 2022/03 - 2022/04

Under Prof. Peter Müller

2022/03 - 2022/04

Worked on devising a methodology for verification and specification of Golang programs that use global variables and package initialization code, using separation logic.

#### TEACHING EXPERIENCE

# • Teaching Assistant

Department of Electrical and Computer Engineering, University of Tehran

- Advanced Programming.

Fall 2020, Spring 2021, Fall 2021

- Data Structures.

Fall 2020

- Design and Analysis of Algorithms.

Spring 2021

- Discrete Mathematics.

Spring 2020, Fall 2020, Spring 2021

- Engineering Probability and Statistics.

Spring 2021

- Operating Systems.

Spring 2022, Fall 2022

## HONORS AND AWARDS

• Stanford School of Engineering (SoE) Fellowship

2024

Awarded a \$12900/quarter stipend and full tuition coverage for one year

Stanford, CA

• Radhia Cousot Young Researcher Best Paper Award 29th Static Analysis Symposium (SAS 2022).

2022/12 Auckland, New Zealand

• Ranked 8th in Regional Contest of ACM-ICPC West Asia Region, Tehran site.

2020

 $\bullet \ Summer@EPFL \ Fellowship \\$ 

Summer 2022

Ranked top 1.5% among 4000 applicants and awarded a 1600CHF/month fellowship over summer.

### **PROJECTS**

• Vampire — 🞧 github.com/vprover/vampire/tree/synthesis-recursive

C++

Implemented a framework within a saturation-based first-order theorem prover for synthesizing recursive programs using structural induction over algebraic datatypes, and superposition calculus.

• Polar — github.com/probing-lab/polar

Python, SymPy

Implemented a polynomial loop-invariant synthesizer for (probabilistic) unsolvable loops, using recurrences.

C++. Z3

Integrated Z3's support for existential/universal quantifiers into the KLEE symbolic execution engine codebase to summarize loops using quantified formulas in first-order logic, and mitigate the path explosion problem.

- Koloocheh —  $\bigcirc$  github.com/daneshvar-amrollahi/Koloocheh

Python, qRPC

A peer-to-peer file-sharing system employs the flooding algorithm for search operations and ensures a small graph diameter by leveraging random graph properties.

## **SKILLS**

## • Programming Languages:

- Experienced in C, C++, Python.
- Familiar with Scala, Go, Bash.
- Tools: cvc5, Z3, KLEE, LATEX.

## **SERVICES**

• Subreviewer: ISSAC 2024