Can Cebeci

can.cebeci@epfl.ch
dslab.epfl.ch/people/cebeci

I'm a 4th year PhD student broadly interested in formal methods and computer systems. My earlier PhD work targets low-effort verification of low-level systems code, with a focus on generic memory models and stable SMT encodings for symbolic execution. Most recently, I have been working on demystifying SMT performance in the context of solver-aided tools, in order to understand and address solver explosion and instability systematically.

Education

École Polytechnique Fédérale de Lausanne, PhD in Computer Science *Sep 2021 – Present*

- Jointly advised by Prof. George Candea (DSLAB) and Prof. Clément Pit-Claudel (SYSTEMF).
- Working on SMT-based automated verification of systems code.

Bilkent University, BS Computer Science. GPA: 3.94/4.0. Ranked 2nd of 165. *Aug 2017 – Jun 2021*

École Polytechnique Fédérale de Lausanne, Exchange. GPA: 5.55/6.0. *Aug 2019 – Feb 2020*

Awards & Honors

- **EPFL IC Distinguished Service Award**, given by the School of Computer and Communication Sciences for exceptional contributions to the PhD program. Received the award both in 2022 and 2023.
- **Rank 7 among more than 830.000 candidates** in the Turkish university placement exam in 2017.
- **"Golden Youth" grant from İşbank**, awarded to 93 Turkish undergraduate freshmen in 2017.
- **Rank 1 in Bilkent and 11 in Turkey** in Google's HashCode in 2018, with a team of five first-year students.

Research Projects

Demystifying SMT explosion and instability *2024 - Present*

(Ongoing) - Developers of SMT-based verifiers strive to optimize and stabilize their SMT encoding, but they lack the tools to do so in an informed fashion. Encoding decision are base, for the most part, on personal intuition, experience and cautionary tales. This project's goal is to develop debugging strategies and tools that help systematically pinpoint the root causes of solver explosion, and abstractions that make the folk wisdom around instability explicit.

Verifying low-level C code with minimal effort *2022 - 2024*

Built TPot, a verifier that targets system components written in C. It supports low-level idioms through a custom, byte-level memory model, and significantly reduces the annotation burden by de-modularizing verification. Instead of modularity, it relies on a tailored specification logic and a bespoke SMT encoding to avoid solver explosion.

Scaling exhaustive symbolic execution *2021*

Experimented with improving the performance of exhaustive symbolic execution in the context of program verification, where bug-finding-oriented search strategies are irrelevant. Incorporated techniques like state merging and incremental SMT solving into KLEE and Angr symbolic executors.

ISA extensions for processing graph applications *2020*

Generated and simulated RISC-V binaries that include special instructions to manage a scratchpad memory. This was done as an undergraduate research project advised by Prof. Özcan Öztürk, in the context of a larger project building a custom processor to efficiently execute vertex-centric iterative graph applications.

Formalization of Amy *2019*

Wrote a pen-and-paper proof of type soundness for a toy subset of Scala used in the compiler construction course at EPFL. This was done within the scope of the same course, given by Prof. Viktor Kuncak.

Publications

Practical Verification of System-Software Components Written in Standard C
Can Cebeci, Yonghao Zou, Diyu Zhou, Clément Pit-Claudel, George Candea. SOSP 2024. artifact

Teaching & Mentorship

- **Supervised internships, theses, semester projects:** *Kevin Solmssen* (C data structure verification), *Priyansh Rathi* (CI pipelines for verifiers), *Stephane Selim* (testing infrastructure-as-code), *Simon Henniger* (state merging in symbolic execution), *Basant Abdelaal* (static pointer analysis for verification), *Yi Rong* (loop summarization in kernel code), and *Loïc Montandon* (Symbolic execution for Python-based smart home configurations).
- **Teaching Assistantships.** *Stanford University (2021-2022, online)* Code in Place. *EPFL (2022-2024)*: Principles of Computer Systems, Software Enterprise, Software Development Project.

Internships

Filemap Inc. - Ankara, Turkey *Jun 2020 – Aug 2020*

Designed and implemented graphical layout algorithms used in a zooming user interface for file systems.

Karel Electronics - Ankara, Turkey *Jun 2019 – Aug 2019*

Built an embedded system to periodically collect and upload audio data from beehives to a remote server.