Anastasios Stefanos Anagnostou

steve-anunknown | Stefanos Anagnostou | anunknown.me | stefanos.anagnostou@outlook.com | +306976227228

Summary

I am a graduate of the School of Electrical and Computer Engineering of the National Technical University of Athens. I enjoy most topics of Computer Science but have a particular interest in Programming Language Theory and Formal Methods. I have had the opportunity to write programs in a variety of settings, ranging from low-level embedded and systems programming, to signal processing and machine learning applications, to automation scripts and high-level functional programming. I am eager to learn new technologies and paradigms and to apply my knowledge in practical settings.

Education

September 2019 - June 2025	Diploma in Electrical and Computer Engineering at the **National Technical University of Athens** (8.45 / 10) Major: Computer Systems, Software and Data Science. Diploma Thesis on Effiecient Equivalence Query Implementation in Active Automata Learning.

Work Experience

Frontistirio Anosi, Teacher September 2021 - June 2023

Taught the “Computer Networks” subject to a class of Greek Professional High School students, in preparation for their university admission exams.

Nokia, Working Student R&D Software Developer June 2024 - December 2024

– I was part of the DevOps team, responsible for automating the building and installation processes of the product. I contributed by enhancing existing scripts in the code base and developing new ones that added further functionality and also participated in the system administration of various servers.

– My main contributions were in the automation of security scans by developing a bash script that interfaced with an API and in the automation of error reporting by developing yet another bash script that reported new errors during the deployment of the product.

Projects

- **Active Automata Learning** - Contributed to the AALpy library for active automata learning in Python by implementing some common but missing equivalence oracles and by helping optimize the implementation of existing ones. Also developed and published my own active automata learning package in Haskell named haal.
- **Blockchain Network and Haskell Development** — Implemented a simple Proof-of-Stake Blockchain network and managed the project using Stack. Used GHC profiling and property-based testing tools and further tested deployment of the project using Docker containers. This project included multithreaded and concurrent programming. My personal website is also written in Haskell.
- **Audio and Computer Vision Algorithms** — Implemented a perceptual audio coding algorithm for signal compression and a variety of computer vision algorithms for detecting interest points in images, detecting moving objects, and classifying videos in different actions.
- **Automation and Data Processing Tools** — Used Bash and Python scripting for automating tasks during the development of projects, performing design-space exploration on multiple experiment parameters, parsing files and gathering data.
- **Compiler Construction and Formal Verification** — Developed a compiler for a Pascal-like programming language in OCaml, using OCamllex and Menhir for the front end and LLVM for the back end. Used Coq in the context of the Advanced Topics on Programming Languages course to study logic and properties of programming languages.
- **Systems Programming and Virtual Machines** — Wrote low-level code for AVR microprocessors, ARM architecture and for the Linux operating system in C. Also developed a stack-based virtual machine with a garbage collector in C, using the GCC language extensions to make it indirectly threaded. Experienced in multithreaded programming, both in C and higher level languages.
- **Algorithm Design and Performance Benchmarking** — Implemented solutions to algorithmic problems in C++ using the STL and wrote driver programs to test and time them against test cases. Also developed and benchmarked CPU branch predictors and synchronization mechanisms.
- **High-Performance and Parallel Computing** — Experienced in optimizing compute-intensive workloads by exploiting parallelism and considering underlying computer architecture, such as cache specifications, NUMA nodes, and OS-level mechanisms like Linux's first-touch policy.

Skills

Languages	Greek (Native), English (Proficient), German (Intermediate)
Programming Languages	Haskell, Python, C, C++, OCaml, Bash, Assembly (ARM, AVR, MIPS)
Programming Paradigms	Concurrent, Parallel, System, Embedded
Tools and Technologies	Git, Docker, Linux