# NANDOR LICKER

n@ndor.email
https://github.com/nandor

## EDUCATION

**University of Cambridge** 2018-2021
*PhD Computer Science*

- Focusing on cross-language optimisations between OCaml and C, supervised by Dr. Timothy Jones
- Supervising undergraduate students in Hardware, Semantics, Compilers and Computer Design
- Received the Vice Chancellor's Scholarship of the Cambridge Trust

**University of Cambridge** 2017-2018
*MPhil Advanced Computer Science, Distinction*

- Received the Cambridge European Scholarship and the Winton Capital Prize for Best Overall Student
- Developed a novel method of verifying the correctness of incremental builds
- Studied subjects related to compilers and computer architectures

**Imperial College London** 2013-2016
*BEng Computing, First-class honours*

- Studied a wide range of subjects, focusing on Compilers, Architectures and Computer Vision
- Worked on a large number of individual and group projects developing optimizing compilers, augmented reality applications, operating systems, games and web applications
- Received the Morgan Stanley Prize, the Fornicary Engineering Prize, the G-Research Prize for Academic Excellence, the Palantir Forward Group Project Prize and the Governor's Prize

## EXPERIENCE

**Apple** Summer 2019
*Software Engineering Intern* *Cupertino, USA*

- Worked on a new clang interpreter embedded into the compiler's frontend to speed up the evaluation of constant expressions at compile time, focusing on constexpr features introduced from C++14 onwards
- Designed a bytecode and a heap layout capable of representing all C++ operations and data structures safely, detecting all possible cases of undefined behaviour, as required by the standard
- Built a code generator and interpreter matching the performance of an existing AST-walking evaluator for toplevel expressions, while speeding up function calls and looping constructs

**OCaml Labs, University of Cambridge** Summer 2018
*Research Intern* *Cambridge, UK*

- Improved the performance of clean builds using the *dune* build system by embedding the OCaml compiler into a process pool managed by the build tool, caching artefacts in memory
- Identified places in the compiler where state is not reset after compilation, preventing multiple compiler invocations in the same process, suggesting fixes to these issues
- Published *opam* packages to manage shared memory and off-heap objects in OCaml

**Palantir** Sprint 2017
*Software Engineering Intern* *London, UK*

· Worked on a micro-service in Java using Elasticsearch, part of a greater ecosystem
· Familiarised myself with Java tooling and development tools

**Stripe** Winter 2017
*Software Engineering Intern* *San Francisco, USA*

· Worked with the Data Platform team, improving the Airflow/Scalding/Redshift infrastructure and implementing incremental snapshotting of production databases in order to significantly reduce costs
· Contributed to internal tools using Go, JavaScript, Java and Scala

**Facebook** Summer 2015 & Summer-Autumn 2016
*Software Engineering Intern* *New York & Seattle, USA*

· Worked with the Compiler Toolchain team on instrumenting x86 binaries and contributed to the LLVM project by creating a late-stage global outlining optimization pass to reduce the size of AArch64 binaries
· Implemented parts of the JavaScript runtime in OCaml for the JSCaml project
· Contributed to the JavaScript interpreter and runtime in the prepack project

**Google** Summer 2014
*Software Engineering Intern* *Zurich, Switzerland*

· Interned with the GeoConsumer Analytics team on the metrics processing pipeline of Google Maps

## PUBLICATIONS

*Detecting Incorrect Build Rules*, N.Licker and A. Rice, ICSE '19, *ACM SIGSOFT Distinguished Paper*