

# Improving the CakeML Verified ML Compiler

Ramana Kumar<sup>1</sup>, Magnus O. Myreen<sup>1,2</sup>, Michael Norrish<sup>3</sup>, and Scott Owens<sup>4</sup>

<sup>1</sup> Computer Laboratory, University of Cambridge, UK

<sup>2</sup> CSE, Chalmers University of Technology, Sweden

<sup>3</sup> Canberra Research Lab, NICTA, Australia

<sup>4</sup> School of Computing, University of Kent, UK

**Abstract.** The CakeML project comprises a mechanised semantics for a subset of Standard ML and a formally verified compiler. We will discuss our plans for improving and applying CakeML in four directions: optimisations, new primitives, a library, and verified applications.

## Introduction

Functional programming languages are often touted as being formal or good for reasoning. While there is some truth to this sentiment in its generality, with CakeML ([cakeml.org](http://cakeml.org)) [1] we seek to push it to its limits and develop a useful ML system that can offer the strongest possible formal guarantees. At the last ML Workshop, we gave an initial presentation about the CakeML language and our verified implementation to date. The overall project has advanced significantly since then; in particular, the compiler has been restructured to better facilitate optimisations, and the definition of the source-language semantics has been improved. We believe that we have arrived at a good baseline for exciting new research. In this talk, we will briefly summarise the new structure and semantics, and then focus on the future directions that we are exploring, with the aim of generating feedback from the ML workshop on our design and planned extensions.

## Topics

**Optimisations:** Recently, the CakeML compiler has been reorganised to be more conventional in its structure, in order to facilitate common optimisations. We hope to discuss how the verified compiler can be improved further, without overly complicating its verification proof.

**Applications:** We are looking for interesting applications where a verified compiler adds real value on the whole. We will explain our plans regarding applications and would like to hear if the ML community has suggestions.

**New primitives:** Many of our planned case studies require some form of new primitives, most notably, some form of verified foreign-function interface. At the time of writing, we are developing such an interface.

**Library:** Much of the value of a verified implementation may reside in the usefulness of its verified libraries. We are considering support for an initially small ‘standard library’ for CakeML programs. To date, CakeML has been designed to be compatible with Standard ML. However, for this library (and other extensions) we might consider not being compatible with Standard ML.

## Summary

The CakeML compiler is being extended and improved in a number of ways. Our talk will discuss the role of verification in applications of ML. We hope to receive advice and feedback from experts in ML compilers on the directions we are taking CakeML.

## References

1. Kumar, R., Myreen, M.O., Norrish, M., Owens, S.: CakeML: A Verified Implementation of ML. In: Principles of Programming Languages (POPL). ACM Press (2014)