# CS 499 – Mechanized Reasoning about Programs

Project Assignment 1

Due date: October 24, 2016

**Abstract**

The goal of the project is to implement a certified compiler for a small language.

## 1 Introduction

One big achievement in certified compilation is the CompCert compiler for a large subset of C towards x86, ARM and PowerPC assemblers [1][1]. This is a huge development: 60,273 lines of specifications, 69,367 lines of proofs, and 11,838 lines of comments for version 2.7.

The compiler developed during the project will be more modest. As a first step, you will study and formalize in Coq the first paper about verified compilation.

In 1967, McCarthy and Painter [2] published a paper entitled "*Correctness of a compiler for arithmetic expressions*". The goal of McCarthy was to obtain machine-checked proofs of correctness for compilers. This work is not machine-checked but, unlike most pen-and-paper work, it is quite detailed.

The input language is a language of arithmetic expressions with only variables, constants and addition. The target language is a small assembly language with load (from memory, from a constant), store, and addition, with an accumulator register. Except for the accumulator that has a specific behavior, all these other "registers" are similar to addresses in main memory. The state in the target language is a total function from addresses to values, and in the source language it is an environment (a mapping from variables to values).

One evaluation function for each language is given, as well as a compilation function. In order to be able to state a theorem about the correctness of this compilation function, it is necessary to be able to relate a state of the source language and a state in the target language. The equivalence of states is done through a mapping between variables and addresses: the value associated to a variable in the source environment and the value associated to the mapped address in the target state should be the same.

The final theorem states that starting from two equivalent states, the evaluation of the compiled code of an expression yields the same result than the evaluation of the expression with the source semantics. The proof is done by induction on the expression.

Despite being very detailed, one hypothesis is missing in the paper: in the equivalence of states, a variable should not be mapped to the accumulator.

## 2 To Do

The goal of this first assignment is to formalize the results of the work of Painter and McCarthy in Coq. To do so, you need of course to read the paper. It is not so easy as the notations are not very close to the notations seen in class.

Then you need to complete the Coq formalization in `McCarthy.v`.

[1] `http://compcert.inria.fr`

# References

[1] X. Leroy. Formal verification of a realistic compiler. *Commun. ACM*, 52(7):107–115, 2009. doi:10.1145/1538788.1538814.

[2] J. McCarthy and J. Painter. Correctness of a compiler for arithmetic expressions. In *Proceedings of Symposium in Applied Mathematics, vol. 19, Mathematical Aspects of Computer Science*, pages 33–41. American Mathematical Society, 1967.