Formal verification of correctness of the Bubble-sort algorithm using Coq

Jamshed Khan
University of Maryland
jamshed@umd.edu

1 Introduction

In this project, we aim to implement the Bubble-sort algorithm with the *Gallina* language embedded in `Coq`'s logic; and then provide a formal verification of the correctness of the implemented algorithm using `Coq`. Since *Gallina* is purely functional, our Bubble-sort implementation will be totally functional too.

2 Motivation

An ideal design to build formally verified correct software using `Coq` consists of writing the program using the embedded *Gallina* language, and then prove the program correct using `Coq`'s proof theory. At this course, we have extensively learned modeling and analysis techniques for programs and programming languages using the `Coq` proof assistant, and how to leverage those tools to mechanically reason about programs. `Coq` provides the necessary frameworks to program formal systems and assisted correctness proving for those. This provides me the motivation to deeply explore the formal verification part, and employ it to the classic Bubble-sort algorithm.

3 Resources

The primary learning materials and other relevant resources are present at *Software Foundations Volume I* [Pierce *et al.*, 2018] and *Verified Functional Algorithms* [Appel,].

References

[Appel,] Andrew W Appel. *Verified Functional Algorithms*.

[Pierce *et al.*, 2018] Benjamin C. Pierce, Arthur Azevedo de Amorim, Chris Casinghino, Marco Gaboardi, Michael Greenberg, Cătălin Hriţcu, Vilhelm Sjöberg, and Brent Yorgey. *Logical Foundations*. Software Foundations series, volume 1. Electronic textbook, May 2018. Version 5.5. http://www.cis.upenn.edu/ bcpierce/sf.