OOPSLA 18, Paper #131 Artifact
Getting Started Guide

Fabian Muehlboeck, Cornell University
Ross Tate, Cornell University
({fabianm,ross}@cs.cornell.edu)

July 2, 2018

1 Requirements

- Coq 8.6 (or compatible - we tested with Coq 8.6 on Windows 7 and 10, and Ubuntu 18.04; conveniently, this seems to be the version that `apt` on Ubuntu is currently installing)
  - On Windows (and apparently OSX), you can download an installer for that version from https://github.com/coq/coq/releases/tag/V8.6
  - On Ubuntu, you can just use `apt`:

    ```
    sudo apt-get update
    sudo apt-get install coq
    ```
- For scripts to work, `cocq` (and, for re-compiling the documentation, if desired, `coqdoc`) must be accessible from `cmd` (on Windows) / `bash` (on Unix-like systems)

2 Running the Artifact

Our artifact is a Coq formalization. To machine-check that formalization, run the script `run.bat` (on Windows) / `run.sh` (on Unix-like systems). The output should at most be echos of the commands that are run (e.g. coqc [filename].v). If no errors show up during this phase (and there should not be any[1]), then all that is left is evaluating the correspondence of the formalization to what we write in the paper, largely following `STARTHERE.html`. For details, see the step-by-step instructions.

Instead of running the scripts, you can also manually compile the files (either from the command line using coqc - you can use the commands from the scripts) or using your favorite Coq IDE, say `coqide`, in the following order:

---

[1] If errors do occur, please check whether you have at least Coq 8.6 installed, using the command `coqc --version`

1. `Common.v`
2. `Tradition.v`
3. `Decide.v`
4. `Convert.v`
5. `Extend.v`
6. `Preprocess.v`
7. `Equate.v`

`Introduction.v` does not contain anything relevant to the proof (it just contains `coqdoc` annotations for the documentation), so it does not need to be compiled. The files `Class.v`, `ClassDistribute.v`, `ClassCompose.v`, and `ClassDisjoint.v` are part of an example of how to use the framework, but excluded from the claims the artifact is meant to support and as such not technically critical for the correctness of the artifact. You can check them anyway by running `run_example.bat` (on Windows) / `run_example.sh` (on Unix-like systems). Since the example depends on the rest of the Coq formalization, running them includes running the core parts of the artifact, too. Finally, if for some reason you want to re-compile the documentation HTML files that are already included, `makedocs.bat` (on Windows) / `makedocs.sh` (on Unix-like systems) re-generates the HTML files using `coqdoc`. In order to do that, it also compiles and thus checks all the Coq files of both the core formalization and the example.