A. Artifact Appendix

A.1 Abstract

Obligatory

A.2 Artifact check-list (meta-information)

- **Program:** The code repsitory for our framework along with the test suite.
- **Compilation:** Python for running the plotting scripts and the Lean4 toolchain, downloaded via `elan`
- **Run-time environment:** Any operating system that supports Docker.
- **Hardware:** Any x86-64 machine.
- **Output:** Key theorems of the paper will be built and shown to have no unsound axioms.
- **How much disk space required (approximately)?:** 10GB
- **How much time is needed to prepare workflow (approximately)?:** 1hr
- **How much time is needed to complete experiments (approximately)?:** 1hr
- **Publicly available?:** Yes
- **Code licenses (if publicly available)?:** MIT
- **Archived (provide DOI)?:**

A.3 Description

A.3.1 Hardware dependencies

None.

A.3.2 Software dependencies

Docker is necessary to run our artifact. the Docker image has dependences needed to compile our framework with Lean4.

A.4 Experiment workflow

Access the docker image `opencompl-ssa` from (`http://doi.org/TODO-TODO-TODO-DOI`), then run:

```
$ docker load -i opencompl-ssa.docker
$ docker run -it siddudruid/opencompl-ssa
# | This clears the build cache,
# | fetches the maths library from the build cache,
# v and builds our framework.
$ cd /code/ssa && lake clean && lake exe cache get && lake build
```

Upon running `make -j4 test`, the test output is printed to `stdout`. The scripts `speedup-time.py` and `speedup-rgn-time.py`, produce PDFs `speedup-time.pdf` and `speedup-rgn-time.pdf` in the directory `/code/lz/test/lambdapure/compile/bench/`:

```
/code/ssa/related-work/alive/compile/bench/speedup-time.pdf
```

To open the pdf file, keep the container running, and in another shell instance, use the `docker cp` command to copy files from within the container out to the host:

```
$ docker container ls # find   ID
$ docker cp <CONTAINERID>:<PATH/INSIDE/CONTAINER> \
            <PATH/OUTSIDE/CONTAINER>
```

For more about `docker cp`, please see: (`https://docs.docker.com/engine/reference/commandline/cp/`)

A.5 Evaluation and expected results

On running `lake build`, the build succeeds with no errors. On applying the path `patch -p1 < axioms.diff` and rebuilding with `lake build`, one should see the axioms that are used by each theorem. Check that these do not use `sorry`, by running `lake build 2>&1 | grep Axioms` and verify by looking that ‘sorry‘ is never on this list.

A.6 Full Workflow Example