A Artifact Appendix

A.1 Abstract

We consider the paper's artifact to be *the included version* of KATER and GENMC, as well as the tools and benchmarks we used in the paper. We stress that the results obtained for the same benchmarks in the future might differ, as the tools might evolve.

As KATER's code is currently undergoing heavy modifications, the included source code of KATER is unstable, and different than the one used in the submitted version of the paper. We will update KATER's code and release a new version of the artifact along with the final version of our paper. We also plan to release KATER on Github.

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

- **Algorithm:** KATER.
- **Program:** KATER, GENMC and C benchmarks.
- **Run-time environment:** Docker.
- **Output:** Console.
- **Experiments:** Scripts that fully reproduce the paper's results are provided.
- **How much disk space required (approximately)?:** ~6 GB.
- **How much time is needed to prepare workflow (approximately)?:** Everything is already set up.
- **How much time is needed to complete experiments (approximately)?:** < 24h (see Section A.7).
- **Publicly available?:** Yes.
- **Code licenses (if publicly available)?:** See GENMC's webpage. For all code in the artifact not belonging to GENMC, GPLv2 applies.
- **Data licenses (if publicly available)?:** GPLv2.
- **Archived (provide DOI)?:** https://doi.org/10.5281/zenodo.7151842.

A.3 Description

A.3.1 How delivered.

The artifact (available on Zenodo) consists of a Docker image containing KATER, a modified version of GENMC that employs KATER-generated consistency checks, and all the benchmarks used in the submitted version of our paper.

A.3.2 Hardware dependencies.

None in particular; having at least 4GB RAM is recommended but not strictly required. Depending on your operating system, Docker might impose some extra hardware restrictions (see Docker's webpage).

A.3.3 Software dependencies.

An operating system on which Docker can be installed (see Docker's webpage) and Docker itself.

Note: Docker often performs poorly in Apple M1 machines. Consider running Docker within a VM in such systems.

A.4 Installation

1. Download and install Docker, in case it is not already installed. On a Debian GNU/Linux distribution, Docker can be installed and started with the following commands:

```
[sudo] apt install docker.io
[sudo] systemctl start docker
```

   We have tested the artifact with Docker 18.09.1 under Debian GNU/Linux.

2. Next, import the Docker container containing KATER:

```
[sudo] docker import kater.docker kater
```

3. Finally, start up the image by issuing:

```
[sudo] docker run -it kater bash
```

A.5 Experiment workflow

The results of the paper are reproduced using bash scripts that run benchmarks which validate our claims.

A.6 Paper claim-list

The most important evaluation claims are summarized below:

1. **Section 7.1, Metatheoretic Results**: KATER can prove various types of metatheoretic results in a few seconds/minutes
2. **Sections 7.2.1, GENMC vs KATER:** KATER is two times slower than GENMC when the latter does not check full consistency, while KATER is two times faster when GENMC checks full consistency.

3. **Section 7.2.2, KATER and memory models:** KATER's consistency checking complexity is proportional to the memory model complexity.
4. **Section 7.2.3, KATER optimization performance:** Saving relations can make KATER two times faster, but never noticeably slower.

Apart from these major claims above, some minor claims regarding the tools' performance in particular benchmarks throughout Section 7. Since these claims easily follow from the tables/graphs reproduced as part of claims 1-4, we do not explicitly list them here.

A.7 Evaluation and expected result

In what follows, we assume that the working directory is `~/kater-benchmarks`. Each subsection below first lists the command(s) that need to be run in order to validate the respective claim, and then provides some comments on the output.

Columns starting with `E:`and `T:` show the number of executions exploredand the time required by the corresponding tool, respectively. For claims 2 and 3, a `*` entry denotes a timeout, while a + entry denotes a failure. As in the paper, we have set a timeout limit of 30m. This limit can be adjusted by modifying L.41 in `get-table-data.sh`.

Note: As the tools have been updated since the submitted version, the obtained results will slightly differ. We expected KATER-generated consistency checks to perform better, though the high-level claims of the paper still hold, and we explicitly mention any major discrepancies below. We also provide a log file with the expected output for each script under `~/kater-benchmarks/logs`, and will update the paper to include the updated results.

A.7.1 Reproducing Claim 1 (< 1h).

```
./get-table1.sh
```

This will run the KATER queries of Table 1 (top-to-bottom, left-to-right). Note that the benchmark names are slightly different than the ones in the paper, and the times reported are different as well. The latter is because of changes that we have made in the tool and the tests. We plan to included the updated results in the revised version of the paper.

A.7.2 Reproducing Claims 2, 3 and 4 (< 6h).

To be able to produce the plots obtained by the results of this step on your host machine, after creating a new empty folder `results` in your host machine, run the docker container using the following command:

```
docker run -v /host/path/to/results:/root/kater-benchmarks/results -it kater bash
```

This will map the host path `/host/path/to/results` to the container's results folder.

Subsequently, you can produce Table 2 and Figures 6,7 with the following commands:

```
./get-table2.sh
./produce-scatters.sh
```

The produced PDF files should appear at the newly created host folder.

For Figure 7, note that the difference between GenMC and KATER in some machines might not be significant. This is because of some optimizations we have made in GenMC so that it performs better even if relations are not saved. We will revise the paper accordingly.

A.8 Experiment customization

A.8.1 Running KATER.

A generic invocation of KATER looks like the following:

```
kater [OPTIONS] <file>
```

To run `kater` in proof mode, simply provide a filename. To run `kater` in code-generation mode, use the `-e` option. Issuing `kater --help` produces a list of all available options.

A.8.2 Running GenMC.

A generic invocation of GenMC looks like the following (use `gater` instead of `genmc` to run the version of GenMC that employs KATER-generated consistency checks):

```
genmc [OPTIONS] -- [CFLAGS] <file>
```

Where `CFLAGS` are options that will be passed directly to the C/C++ compiler, and `OPTIONS` include several options that can be passed to GenMC (`genmc --help` prints a full list). `file` should be a C/C++ file that uses `pthreads` for concurrency.

More information regarding the usage of GenMC (as well usage examples) can be found at GenMC's manual.

A.8.3 Available benchmarks.

The C benchmarks we used for our paper are located under `~/kater-benchmarks/benchmarks`. KATER models can be found under `~/kater/kat`, and KATER tests under `~/kater/tests`. GENMC tests can be found at GENMC's repository, a modified local copy of which can be found at `~/genmc-tool`.

In the above repository under `tests` (and the relevant sub-directories), there is a separate folder for each benchmark, that contains the "core" of the test case, as well as the expected results for the test case, some arguments necessary for the test case to run, etc. In order to actually run a test case, one can run the tool with one of the test case variants, which are located in a folder named 'variants', in turn located within the respective test case folder.

For example, to run a simple store buffering test with GENMC, please issue:

```
genmc ~/genmc-tool/tests/correct/litmus/SB/variants/sb0.c
```

A.8.4 Code Layout.

KATER's source code is located at `~/kater/src`.

GENMC's source code is located at `~/genmc-tool/src`. KATER-generated consistency checks can seen at `RC11Checker.cpp`. A more detailed explanation of GENMC's code layout can be found at this paper.