

# The precision benchmarks and evaluation

---

This folder contains all scripts and necessary folders to gather evaluation results.

## File structure

The following are the components of the artifact:

- **benchmarks**: Contains all benchmarks mentioned in the paper.
- **configs**: Contains all configurations files for each tool.
  - **clam.yaml**: the yaml file for running Clam/Crab.
    - Basically, it sets Crab to use Zones and delays widening by 1.
  - **mopsa\_rel.json**: the json file for running Mopsa.
    - Use relational domain, the command line will select Octagons.
  - **mopsa\_int.json**: another file for running Mopsa using nonrelational domain.
- **data**: Stores all csv files collected by script **scripts/get\_assert\_results.py**.
- **outputs**: Includes the outputs made by each tool.
  - **rgn**: the results from the summarization domain.
  - **obj**: the results from the MRU domain.
  - **mopsa**: the results from the Mopsa recency domain.
- **paper\_results**: Shows the results mentioned in the paper.
  - We do not directly generate a LaTeX version. The results mentioned in the paper will be printed as a log file.
- **scripts**: Keeps all scripts for running experiments.
  - **clean\_up.sh**: Clean up the results in the previous folders.
  - **get\_assert\_results.py**: Dump assertion statistics from the **outputs** folder into **data/\* .csv** files.
  - **get\_paper\_results.py**: Generate the paper table based on the **data/\* .csv** results.
  - **run\_experiment.sh**: Runs evaluation for one abstract domain.
  - **run\_precision\_experiments.sh**: Runs all evaluation for all domains.

## The C code

Each C code contains several checks, including assertions or memory safety checks.

Assertion: **assert(length <= capacity);**

### Crab

It will print detailed invariants in **outputs/<domain\_name>/<test\_case>/code.crabir**.

- If the assert check is SAFE, it will show a comment like this:

```
// loc(file=benchmarks/byte_buf.c line=63 col=3) id=1 Result: OK
```

- If the check is Warning:

```
// loc(file=benchmarks/byte_buf.c line=63 col=3) id=1 Result: FAIL -- num of warnings=1
```

### Mopsa

It will dump a json file in **outputs/mopsa/<test\_case>/log.json**.

- If the assert check is SAFE, Mopsa does not show it in json file.
- If the assert is WARNING, Mopsa will have a warning report with kind -- **Assertion failure**.

IsDerefCheck: `ASSERT_IS_DEREF(h->cookie, idx);`

## Crab

We turn check to assertion. So it will show the same result as assertion.

## Mopsa

We use Mopsa stub `_mopsa_assert_valid_bytes`:

The ``_mopsa_assert_valid_bytes`` built-in is defined as a contract using the bytes and offsets functions to access pointer meta-information.

Ref: <https://mopsa.gitlab.io/mopsa-analyzer/user-manual/stubs/c.html>

- If the assert is WARNING, Mopsa will give a warning with kind - `Stub condition`.

By default, Mopsa does not print invariants. If you want to view invariants at a specific line of code, please refer to [this page](#).