Reproducibility Checklist for *Learning from Answer Sets via Single-Shot Disjunctive ASP Encoding*

1 Environment Setup

1.1 Clingo

The latest version of `clingo` can be downloaded at `https://github.com/potassco/clingo/releases`. We used version 5.6.2.

1.2 LASCO

The latest version of LASCO can be downloaded following the instructions on the `README` file of the present repository. We summarize here the main steps:

1. Download the present repository
2. Download the latest version of `ghc` from `https://www.haskell.org/ghc/`. We used version 9.4.8.
3. Download the latest version of `cabal` from `https://www.haskell.org/cabal/`. We used version 3.14.1.1.
4. Go to the `code` directory and compile using the command `cabal build`
5. Extract the compiled binary from `dist-newstyle/build/.../lasco`

1.3 ILASP

The latest version of ILASP can be obtained from `https://github.com/ilaspltd/ILASP-releases/releases`. We used version 4.4.0.

2 Datasets

The instances of the presented tasks can be downloaded from the current repository in the directories `aaai26/table1_instances/` and `aaai26/table2_instances/`. Notice that instances in `aaai26/table2_instances/` were originally presented in *Towards ILP-Based LTLf Passive Learning* (Ielo, Law, Fionda, Ricca, De Giacomo, Russo, 2023). The original repository containing those instances is publicly available at `https://github.com/ilp2023-27/data`.

3 Evaluation

We used the command `time` to measure the running times and we used the command `timeout` to cut the computation after one hour.

Let `inst` be a test instance. We show the command to solve `inst` on the presented configurations.

ILASP2i

```
/usr/bin/time -f "%e" timeout 60m ILASP --version=2i inst
```

ILASP4

```
/usr/bin/time -f "%e" timeout 60m ILASP --version=4 inst
```

LASCO1t

```
/usr/bin/time -f "%e" timeout 60m lasco -i inst
--solve clingo -o inst.enc --solve-mode opt -t 1
```

LASCO2t

```
/usr/bin/time -f "%e" timeout 60m lasco -i inst
--solve clingo -o inst.enc --solve-mode opt -t 2
```

LASCO4t

```
/usr/bin/time -f "%e" timeout 60m lasco -i inst
--solve clingo -o inst.enc --solve-mode opt -t 4
```

LASCO8t

```
/usr/bin/time -f "%e" timeout 60m lasco -i inst
--solve clingo -o inst.enc --solve-mode opt -t 8
```

LASCO16t

```
/usr/bin/time -f "%e" timeout 60m lasco -i inst
--solve clingo -o inst.enc --solve-mode opt -t 16
```

LASCO16tf

```
/usr/bin/time -f "%e" timeout 60m lasco -i inst
--solve clingo -o inst.enc --solve-mode first -t 16
```

4 Used Hardware

All tests were run on an Ubuntu 24.04 desktop machine with a 13th Gen Intel(R) Core(TM) i7-13700KF (24 threads) and 16 GB RAM.

5 Additional Notes on Reproducibility

- **Randomness.** The evaluation pipeline does not involve any stochastic component. All tools (LASCO, ILASP, Clingo) produce deterministic results for fixed inputs. Therefore, no random seed needs to be set.

- **Timing variability.** Although the logic of the computation is deterministic, running times may vary due to internal optimizations and heuristics used by Clingo. In the experiments reported in the paper, we measured execution time using the `/usr/bin/time` utility, with a 60-minute timeout. For more accurate and robust results, we recommend repeating each experiment 3 times and reporting the average runtime.

- **Reproducing paper tables.** The experimental commands listed in Section 3 exactly correspond to those used to generate Tables 1 and 2 in the main paper. Each row of those tables corresponds to running a specific configuration (e.g., LASCO4t or ILASP2i) on a particular test instance.

- **Repository structure.** The datasets used for Table 1 and 2 are included in the repository under the directories `aaai26/table1_instances/` and `aaai26/table2_instances/`.

- **Documentation.** The repository contains a `README.md` file detailing installation, compilation, and usage instructions, including a summary of all dependencies and requirements.