ARTIFACT DESCRIPTION

A. Abstract

The provided package includes all the examples mentioned in the paper coded (mostly) in MATLAB and in the UPPAAL tool. All the example codes are provided in separate files to simplify their use. To obtain the results of Example IV.1 and Example V.1 (using MATLAB) and Table III (using MATLAB and UPPAAL), users have to run the provided codes which produce the output files and figures as described in this section. The users are welcome to change the input parameters to analyze different case studies.

B. Artifact checklist

- **Algorithm:** FAn method that is proposed in this paper, Brute Force (BF) search method, Time Automata (TA) based method.
- **Program:** Matlab (for FAn and BF methods), UPPAAL (for TA method)
- **Output:** Fig.1, Fig.2(c-e), results of Table III.
- **Experiment workflow:** Install MATLAB/Octave; Install UPPAAL; Unzip the submitted package; Run MATLAB code for each experiment; Open .xml files with UPPAAL and verify the queries;
- **System requirements:** Java 7 (and later) and MATLAB (2007 and later) must be installed in the user's system which is equipped with Linux or Windows (XP and later).

C. Download AE package

The users can clone or download the AE package from `https://github.com/behrouzian/Amir-Behrouzian.git`. In Linux, the package is cloned as the followings.

```
$ cd /a directory to deploy the zip file/
$ git clone https://github.com/behrouzian/
    Amir-Behrouzian.git
```

D. Example 4.1 and 5.1

Running the files `"AE_Behrouzian\Example41.m"` and `"AE_Behrouzian\Example51.m"` provides results for Example IV.1 and Example V.1 and generates Fig.1 and Fig.2, respectively. The run-time of programs are reported in `"Command Window"`.

E. Table 3

The folder `"AE_Behrouzian\Table3\"` contains the implementation of the experiments mentioned in Table III. In the table, the run-time of three example using the three methods FAn, BF and TA are compared. Running the files in `"AE_Behrouzian\Table3\BF_method\"` and `"AE_Behrouzian\Table3\FAn_method\"` in MATLAB reports the run-time and results the number of Deadline Hits (DHs) m using BF and FAn methods, respectively.

UPPAAL is a Java-based free software whose installation files are provided in `"AE_Behrouzian\Install_UPPAAL"` folder in the package submitted. `"AE_Behrouzian\Table3\TA_method\"` contains the UPPAAL-importable timed-automata models of the experiments mentioned in Table III.

To run the UPPAAL verifications, the following steps needs to be taken. UPPAAL must be installed first. In windows, double click on `"AE_Behrouzian\UPPAAL\uppaal.jar"`. In Linux, follow the steps below.

```
$ cd /path/to/UPPAAL/
$ java -jar uppaal.jar
```

In case of `"permission denied"` error, the proper permission must be given to the folder `"AE_Behrouzian\UPPAAL\bin-Linux"` as followings.

```
$ chmod 755 bin-Linux/*
$ java -jar uppaal.jar
```

Once UPPAAL is loaded (in both Windows and Linux), from `"File>Open System"` open the provided timed-automata models for each example in `"AE_Behrouzian\Table3\TA_method\"`. In the `"Verifier"` tab, click on the property `"sup : DH"` in the `"Overview"` box and then click `"Check"` to verify the maximum possible number of DHs. The run-time of the verification is reported in the `"Status"` box as `"verification elapsed time"`. All the three experiments use the same model with different input parameters which are specified in `"System declarations"`.