This folder contains the PRISM model described in the paper

"Quantitative Programming and Continuous-Time Markov Chains"

Author: Eneia Nicolae Todoran (SYNASC 2023 submission 26)

- PRISM is a widely used probabilitic model checking tool

www.prismmodelchecker.org

- This file describes two experiments are based on
  - The PRISM model contained in file `epollingsystem2-ctmc.prism`
  - The PRISM properties contained in file `epollingsystem2-ctmc.props`

- In the experiments presented below `vs1` is the PRISM counterpart of $\overline{\mathcal{L}}_{QP}$ variable v_{s1} (declared in module $S1$), and `aserve1` is the PRISM counterpart of action label a_{serve1}, from the $\overline{\mathcal{L}}_{QP}$ example program presented in Section III (further explanations regarding the meaning of PRISM variable names employed below are provided in the paper).
  - The experiments presented below were performed using the following options (available from the PRISM GUI): "linear equations method" = Gauss-Seidel (Jacobi, the default method, does not converge for some experiments), and "Termination max. iterations" = 100000 (the default limit is 10000).

- Due to space limitations, the experiments presented below are not presented in the paper.

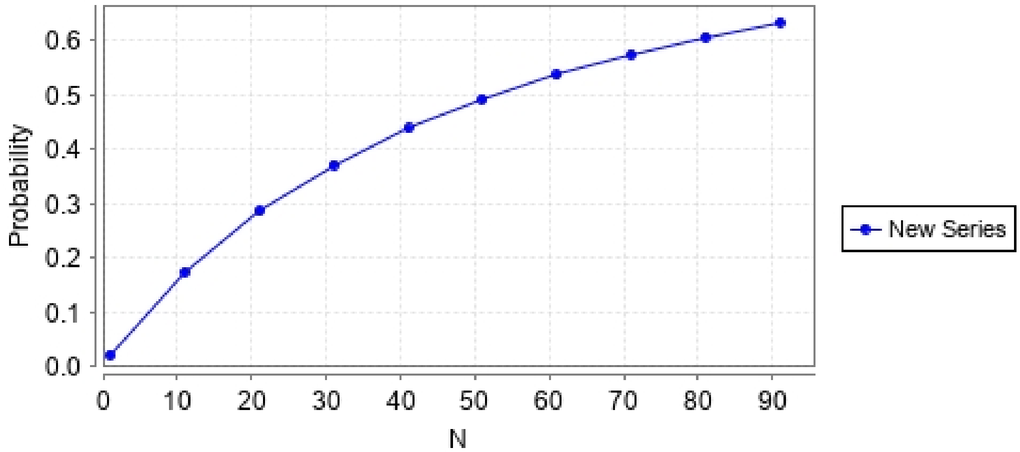

Figure 1: The probability that in the long run station $S1$ is idle

- In the PRISM experiments presented below we put $\nu = 25$.
- We use the following constants:

```
const int nmax = 25;
const int N;
const int T;
```

The execution rate of the statement $\langle w_1 := \ \mathsf{delay}(20, \mathsf{rand}(\nu)), z_{delay1}\rangle$ is calculated as the inverse of the duration expression z_{delay1}.[1] The duration expression z_{delay1} and the corresponding execution rate are fixed (they do not depend upon the current state). The $\overline{\lambda}_{QP}$ primitive $\mathsf{delay}(20, \cdots)$ delays the execution of $\mathsf{rand}(\nu)$ by exactly 20 time units (20 elementary evaluation steps) and the evaluation of primitive $\mathsf{rand}(\nu)$ is also fixed, requiring 4 time units (in total 24 time units).[2] Hence, in the PRISM experiments presented in the paper the corresponding CTMC rate is defined as follows: `formula zdelay1 = (1/24)`. In the PRISM model contained in file `epollingsystem2-ctmc.prism` it is considered a different rate `zdelay1` that corresponds to the $\overline{\mathcal{L}}_{QP}$ statement $\langle w_1 := \ \mathsf{delay}(20 * N, \mathsf{rand}(\nu)), z_{delay1}\rangle$, namely:

```
formula zdelay1 = 1/((20*N)+4)
```

1. Using the PRISM property specification language, we can specify the probability that in the long run station $S1$ is idle as follows:

   ```
   S=? [ vs1=0 ]
   ```

   Figure 1 presents a PRISM experiment for this property. In this experiment `N` ranges from `1` to `100` with step `10`.

[1] This statement is in the module $S1$ presented in Section III of the paper.
[2] $\overline{\lambda}_{QP}$ is the functional sub-language of the language $\overline{\mathcal{L}}_{QP}$ presented in the paper.

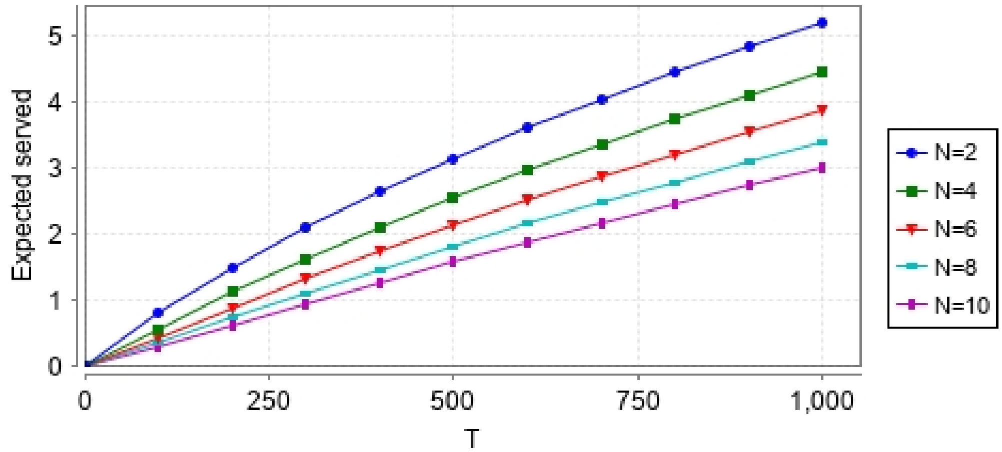

Figure 2: Expected number of times station $S1$ is served

2. For the next experiment we define a rewards structure `"served"`

```
rewards "served"
 [aserve1] true : 1;
endrewards
```

We can compute the expected reward (number of times station $S1$ is served) accumulated by time `T` as follows:

```
R{"served"}=? [C<=T]
```

In the experiment presented in Figure 2, `N` ranges from `2` to `10` with step `2`, and `T` ranges from `0` to `1000` with step `100`.