Interactive Markov Chains Analyzer

Dennis Guck

Contents

1 IMCA's Input Files **1**
1.1 The .imc and .ipc File Format 1

2 Functionality of IMCA **2**
2.1 Minimization of an IMC 2
2.2 Time- and Interval-Bounded Reachability 2
2.3 Unbounded Reachability 3
2.4 Expected Time and Expected Steps 4
2.5 Scheduler Synthesis 4
2.6 Plot Function 4
2.7 Log-File 4

The Interactive Markov Chains Analyzer (IMCA) is able to analyze IMC's and IPC's. It is written in C++ and uses the following libraries:

- GMP [1]
- MPFR [2]
- MPFRCPP [3]
- QT 4.6 [4]

1 IMCA's Input Files

IMCA expect two input files: a .imc-file describes an IMC with its Markovian and interactive transitions and initial and goal states, and a .ipc-file describes an IPC with its probabilistic and interactive transitions and initial and goal states. After opening of a new IMC or IPC, the file is displayed in the model view window.

1.1 The `.imc` and `.ipc` File Format

The file format contains in the first row the initial states. If there are several initial states, each state is seperated by a space character, and // will end the line. The second row contains the goal states. There are the same constraints as for the initial row. The following rows describe the transitions of the IMC or rather IPC. One transition is described as follows:

src dst prob/rate/action

where src is the source state, dst the destination state, and prob/rate/action the probability, rate or action to move from src to dst.

Listing 1: Example .imc-file

```
s0 // initials
s3 // goals
s0 s1 a
s0 s3 b
s1 s3 0.5
s1 s2 0.5
s2 s1 0.25
s2 s2 0.5
s2 s3 0.25
s3 s2 c
```

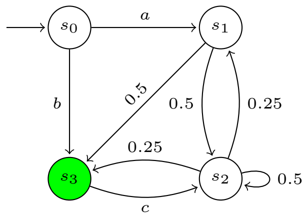

Figure 1: Associated IMC to Listing 1.

2 Functionality of IMCA

The main functionalities are to calculate the maximum and minimum time-bounded reachability probability [9], expected time, and expected steps. Besides it is possible to extract the scheduler for the time-bounded reachability probability. Moreover, it is possible to minimize an IMC via bisimulation.

2.1 Minimization of an IMC

The computation time depends for every computation on the complexity of the model. Because of that, it is implemented a minimization algorithm, which works with a bisimulation algorithm described in [5]. A condition is, that the IMC is closed. However, the program make it closed and well formed. After the minimization is finished, the minimized model is displayed in the model view.

2.2 Time- and Interval-Bounded Reachability

It is possible to compute the time- and the interval-bounded reachability probability. The theory is explained in [9]. The inputs that are needed for the computation are:

- error
- lower−bound
- upper−bound
- interval

The error will specify the error-bound ϵ. The lower- and upper-bound specify the interval for which we calculate the reachability probability. If the lower-bound is 0, we calculate the time-bounded reachability, otherwise the interval-bounded. The variable interval have to be specified to record the progress of the probability during the computation within the defined interval. Furthermore, we have to choose between the computation of the maximum and minimum time-bounded reachability.
Another feature is to start the computation with an advanced error bound [8]. This option allows a faster computation. A comparison between the advanced and normal error-bound, with the workstation cluster from [7] with one workstation per cluster, is given in Table 1 and 2.
Since we calculate everything on the discretized IMC, we also can compute the step-bounded and step-interval-bounded reachability for IPC's.

Time Bound z	error bound ϵ	iteration steps N	probability
1	10^{-2}	173	$3.07535 \cdot 10^{-6}$
1	10^{-4}	17323	$3.06783 \cdot 10^{-6}$
1	10^{-6}	1732390	$3.06795 \cdot 10^{-6}$
10	10^{-2}	200	$7.1097 \cdot 10^{-5}$
10	10^{-4}	20026	$7.10122 \cdot 10^{-5}$
10	10^{-6}	2002700	$7.10156 \cdot 10^{-5}$
100	10^{-2}	200	0.000881425

Table 1: value-iteration with advanced error bound

Time Bound z	error bound ϵ	iteration steps N	probability
1	10^{-2}	201	$3.06691 \cdot 10^{-6}$
1	10^{-4}	20055	$3.06807 \cdot 10^{-6}$
1	10^{-6}	2005400	$3.06795 \cdot 10^{-6}$
10	10^{-2}	20055	$7.10174 \cdot 10^{-5}$
10	10^{-4}	2005400	$7.10156 \cdot 10^{-5}$
10	10^{-6}	200540366	aborted
100	10^{-2}	2005400	0.000882816

Table 2: value-iteration with normal error bound

2.3 Unbounded Reachability

The unbounded reachability probability will be calculated on the embedded IPC. We use the value iteration algorithm from the time-bounded reachability and iterate until we reach a fixpoint. The fixpoint is a point where the unbounded reachability probability for all states do not change after k iteration steps. Hence

$$\forall s \in S. \forall i > k. p_{i-1}(s) = p_i(s). \tag{1}$$

The fixpoint is reached, if equation 1 is satisfied for 1000 steps, hence for $i = k+1, \ldots, k+1000$. Besides it is possible to accept a variance ϵ to satisfy the fixpoint ($|p_{i-1}(s) - p_i(s)| = \epsilon$). Therefore exists the input field error. As before it is possible to choose between the maximum and minimum unbounded reachability prbability.
Since we calculate the unbounded reachability on the embedded IPC, we also can compute the unbounded reachability for IPC's.

2.4 Expected Time and Expected Steps

Another function is to obtain the expected time or the expected steps to reach a set of goal states. The expected time and expected steps will also be calculated with a value iteration algorithm, and iterate until a fixpoint is reached, as described in 2.3. Further it is possible to choose between the maximum and the minimum.
In case of an IPC, it is only possible to calculate the expected steps.

2.5 Scheduler Synthesis

The scheduler synthesis is implemented for the maximum time- and interval-bounded reachability 2.2. The theory is the same as in [6]. Therefore, for each step in the value iteration, we save the chosen action for each interactive state. After the computaton is finished, the scheduler is displayed in the log-file (c.f. Section 2.7). The same principle is applicable for the expected time and expected steps, but is not featured in the tool yet.

2.6 Plot Function

The plot function is supported for the time- and interval-bounded reachability (c.f. section 2.2). Therefore, we use the input of the interval field, that describes how often we capture the probability during the computation. After the computation, it is possible to choose a state $s \in S$ and plot the developing of the probability of this state for the given interval [lower-bound;upper-bound]. Further, it is possible to save the plot.

2.7 Log-File

The log-file registers the open model, the chosen computation with their inputs, the computation time, the result for the initial state and additionally the scheduler. Besides, the computation steps are displayed in the log-file. The log-file can also be saved.

Listing 2: Example log-file for time-bounded reachability of the IMC in figure 1.1

```
Logfile
========================
IMC Analyzer
========================
```

```
open file: "../early.imc"

States: 4
Initial States: 1
Goal States: 1
Transitions: 8

maximum time-bounded reachability probability
making goal states absorbing (time-bounded reachability)...
closing imc...
making deadlock states absorbing...
discretizing imc...
time interval:       [0,1]
interval time:       0.1
error bound:        1e-06
max exit rate:      1
step duration:      2e-06
step interval:       [0,500001]
interval step:       50000
computation start...
computation end...
Time: 1.92002 seconds
initial state prob:
s0 = 1
```

References

[1] The GNU Multiple Precision Arithmetic Library. http://gmplib.org/.

[2] The MPFR Library. http://www.mpfr.org/.

[3] The Multiple Precision Floating-Point Reliable Library C++ Interface. http://beshenov.ru/mpfrcpp/.

[4] Qt. http://doc.qt.nokia.com/4.6/index.html.

[5] C. Baier and J.-P. Katoen. *Principles of Model Checking*. The MIT Press, May 2008.

[6] D. Guck. Analysis and scheduler synthesis of time-bounded reachability in continuous-time Markov decision processes. *Bachelorarbeit, RWTH Aachen University*, 2010.

[7] B. R. Haverkort, H. Hermanns, and J.-P. Katoen. On the use of model checking techniques for dependability evaluation. *Reliable Distributed Systems, IEEE Symposium on*, 0:228, 2000.

[8] J.-P. K. Taolue Chen, Tingting Han and A. Mereacre. Computing Maximum Reachability Probabilities in Markovian Timed Automata. *Technical Report AIB-2010-06*, 2010.

[9] L. Zhang and M. Neuhäußer. Model Checking Interactive Markov Chains. *Lecture Notes in Computer Science*, 6015:53–68, 2010.