Erlang Stages

AVACS S3
Phase 2

July 28, 2011

1 Description of the Model

In this test case, we considered an example taken from [5], which is shown in Figure 1. Unlike the original example, we modeled the process as a Continuous Time Markov Decision Process (CTMDP) instead of an Interactive Markov Chain (IMC). Former interactive actions are transformed into tuples (r, a), where r is a rate and a is an interactive action. These kind of actions are resolved by statically defining a set $D = \times_{s=1}^{n} D_s$ of *decisions* where D_s is a finite set of decisions that can be taken in state $s \in S$. For each decision vector d, a rate Matrix Q^d is defined. States $s_{2,1}, \ldots, s_{2,30}$ represent

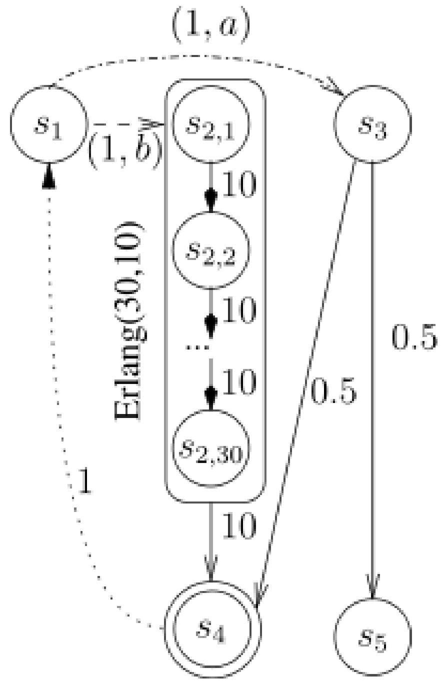

Figure 1: CTMDP with an Erlang(30,10)-Transition

an unfolded Erlang(30,10) transition, i.e. 30 consecutive Markovian transitions each of which has rate 10. In this test case, we are interested in checking whether:

1. state s_4 is reached from state s_1 **within** T **time units** with probability $< x$
2. state s_4 is reached from state s_1 **in the interval** $[t_0, T]$ with probability $< x$.

First, we check the more concrete property corresponding to the CSL formula $P_{<x}(F^{[0,T]}s_4)$ and go on to the general case corresponding to $P_{<x}(F^{[t_0,T]}s_4)$. We consider a single atomic proposition s_4 which holds only in state s_4.

We implemented our model checking algorithm [2] in an extension of the probabilistic model checker MRMC [4]. In addition, we implemented a method to compute long-run average state probabilities [3]. The implementation is written in C, using sparse matrices. Parallelism is not exploited. All experiments are performed on an Intel Core 2 Duo P9600 with 2.66 GHz and 4 GB of RAM running on Linux.

2 Results

To compute the results for $P_{<x}(F^{[0,T]}s_4)$, state s_4 is made absorbing by removing the transition from s_4 to s_1 (shown as a dashed line in the figure), as discussed in Subsection 4.3 in [2]. Table 1 contains the results and efforts to compute the maximal reachability probabilities for $T = 4$ and 7 with the adaptive and non-adaptive variant of the uniformization approach. The time usage is given in seconds. It can be seen that the adaptive version is much more efficient and should be the method of choice in this example. The value of ϵ that is required to prove $P_{<x}(F^{[0,T]}s_4)$ depends on x. E.g., if $T = 4$ and $x = 0.672$, $\epsilon = 10^{-4}$ is sufficient whereas $\epsilon = 10^{-3}$ would not allow one to prove or disprove the property. Columns "lower" and "upper" represent the bounds of the optimal policy. Our algorithm (Algorithm 2 in [2]) relies on these given bounds although it would be possible to start with an exactly known probability vector.

To compute the result for $P_{<x}(F^{[t_0,T]}s_4)$, the two step approach is used. We consider the Interval $[3, 7]$. Thus, in a first step the optimal gain vector at time 3, $\mathbf{a}_{[3,7]}$ is computed from the CTMDP where s_4 is made absorbing. Then, the resulting upper and lower bounds, $\overline{\mathbf{g}}_3$ and $\underline{\mathbf{g}}_3$, of the optimal policy are used as terminal conditions to compute upper and lower bounds of the optimal policy at time 0 from the original process including the transition between s_4 and s_1. Apart from the final error bound ϵ for the spread between $\underline{\mathbf{g}}_0$ and $\overline{\mathbf{g}}_0$, an additional error bound ϵ_1 $(< \epsilon)$ has to be defined which defines the spread between $\underline{\mathbf{g}}_3$ and $\overline{\mathbf{g}}_3$.

Table 2 includes some results for different values of ϵ and ϵ_1. The columns headed with $iter_i$ $(i = 1, 2)$ contain the number of iterations of the i-th phase. It can be seen that for this example, the first phase requires more effort such that ϵ_1 should be chosen only slightly smaller than ϵ to reduce the overall number of iterations. Note that, in this case, it is important to take time-dependent policies to arrive at truly maximal reachability probabilites. The maximal value obtainable for time-abstract policies (using a recent algorithm for CTMDPs [1, 4]) is 0.584284 (versus 0.6717787) for a time bound of 4.0, and 0.9784889 (versus 0.9828449) for a time bound of 7.0.

T	ϵ	Uniformization $K = 5$				
		lower	upper	steps	iter	time (s)
4.0	10^{-3}	0.671006	0.672001	143	715	0.01
4.0	10^{-4}	0.671701	0.671801	720	3600	0.03
4.0	10^{-5}	0.671771	0.671781	5921	29605	0.10
4.0	10^{-6}	0.671778	0.671779	56361	281805	0.87
7.0	10^{-3}	0.981858	0.982855	268	1340	0.02
7.0	10^{-4}	0.982746	0.982846	1283	6415	0.04
7.0	10^{-5}	0.982835	0.982845	10350	51750	0.22
7.0	10^{-6}	0.982844	0.982845	97268	486340	1.64

T	ϵ	Uniformization $K_{max=20}, \omega = 0.1$				
		lower	upper	steps	iter	time (s)
4.0	10^{-3}	0.671083	0.672022	29	201	0.01
4.0	10^{-4}	0.671772	0.671803	211	774	0.02
4.0	10^{-5}	0.671778	0.671781	2002	5038	0.09
4.0	10^{-6}	0.671778	0.671779	19473	401031	0.63
7.0	10^{-3}	0.982753	0.982852	50	341	0.02
7.0	10^{-4}	0.982836	0.982846	364	1333	0.04
7.0	10^{-5}	0.982844	0.982845	3463	8098	0.19
7.0	10^{-6}	0.982845	0.982845	33747	68876	1.50

Table 1: Bounds for Reaching s_4 in $[0, T]$, i.e. $P^{max}_{s_1}(F^{[0,T]} s_4)$

ϵ_1	ϵ = 1.0E-03			
	time	bounded prob.	$iter_1$	$iter_2$
9.0E-4	0.97170	0.97186	207	90
5.0E-4	0.97172	0.97186	270	89
1.0E-4	0.97175	0.97185	774	88
1.0E-5	0.07175	0.97185	5038	88

ϵ_1	ϵ = 6.0E-04			
	time	bounded prob.	$iter_1$	$iter_2$
9.0E-4	-	-	-	-
5.0E-4	0.97176	0.97185	270	93
1.0E-4	0.97178	0.97185	774	91
1.0E-5	0.97179	0.97185	5038	91

Table 2: Bounds for Reaching s_4 in $[3, 7]$, i.e. $P^{max}_{s_1}(F^{[3,7]} s_4)$

References

[1] Tomás Brázdil, Vojtech Forejt, Jan Krcal, Jan Kretínský, and Antonín Kucera. Continuous-time stochastic games with time-bounded reachability. In *FSTTCS*, volume 4 of *LIPIcs*, pages 61–72, 2009.

[2] Peter Buchholz, Ernst Moritz Hahn, Holger Hermanns, and Lijun Zhang. Model checking algorithms for ctmdps. In *CAV*, 2011.

[3] Ronald A. Howard. *Dynamic Programming and Markov Processes.* MIT Press, Cambridge, Massachusetts, 1960.

[4] Joost-Pieter Katoen, Ivan S. Zapreev, Ernst Moritz Hahn, Holger Hermanns, and David N. Jansen. The ins and outs of the probabilistic model checker MRMC. In *QEST*, pages 167–176, 2009.

[5] Lijun Zhang and Martin R. Neuhäußer. Model checking interactive Markov chains. In *TACAS*, pages 53–68, 2010.