FlyFast: **A Mean Field Model Checker**

Diego Latella[1], Michele Loreti[2,3(✉)], and Mieke Massink[1]

[1] Consiglio Nazionale delle Ricerche - Istituto di Scienza e Tecnologie dell'Informazione 'A. Faedo', CNR, Pisa, Italy
{diego.latella,mieke.massink}@isti.cnr.it
[2] Università di Firenze, Florence, Italy
michele.loreti@unifi.it
[3] IMT Alti Studi, Lucca, Italy

Abstract. We present FlyFast, a recently introduced on-the-fly *mean field* model checker for the verification of time-dependent probabilistic properties of individual objects in the context of large populations. An example of its use is illustrated analysing a push-pull gossip protocol. Such protocols form the basis on top of which many smart collective adaptive systems are built. Typical properties are the replication of a fresh data element throughout a network, the coverage of the network, and the time to convergence.

Keywords: Mean field model checking · Collective Adaptive Systems · Discrete time markov chains · Self-organisation · Gossip protocols

1 Introduction

FlyFast is a, first of its kind, *on-the-fly mean field* probabilistic model checker. Its purpose is the automatic verification of bounded PCTL (Probabilistic Computation Tree Logic) properties of a *selected individual* in the context of systems that consist of a *large number* of (similar, but) independent, *interacting objects*. Typical examples of such systems are large scale Collective Adaptive Systems (CAS) and distributed algorithms for sharing data in a distributed network, such as gossip protocols. Following the mean field approach proposed in [7], an on-the-fly mean field model checking algorithm was developed and proven correct in [4,6]. Models that can be analysed by FlyFast are time-synchronous DTMC-based population models in which each object performs a probabilistic step in each discrete time unit, moving between its local states and possibly returning to the same state. Objects interact in an indirect way, via the global state of the system. In fact, the evolution of the global system is specified by the *local* transition probabilities of an object. The latter are the same for each object in the population (i.e. one abstracts from their identity) and may depend on the distribution of local states of all objects in the system, i.e. its occupancy

measure vector[1]. When the number of objects is large (at least several hundreds) the overall behaviour, in terms of its occupancy measure vector, can be approximated by the deterministic solution of a difference equation, which is called the 'mean field' [7]. This iterative approach to obtain the occupancy measure vector has shown to combine very well with an on-the-fly probabilistic model checking approach [6]. The latter is parametric w.r.t. the semantics interpretation of the model specification language and in FlyFast it is instantiated on a mean-field population semantics. The algorithm consists of two phases, an expansion phase and a computation phase. Both phases are linear in the number of states and transitions of the expansion of the initial state of the selected object and occupancy measure vector [6] for the time bounded fragment of PCTL. FlyFast has been applied on a.o. bike sharing [6], client-server systems and computer worm epidemic models [5].

FlyFast is provided within the jSAM (java StochAstic Model Checker) framework which is an open source Eclipse plugin[2] integrating a set of tools for stochastic analysis of concurrent and distributed systems specified using process algebras. We illustrate the use of FlyFast using a push-pull gossip protocol as a running example [1,2]. Gossip protocols provide a scalable, simple, robust and fully decentralised communication mechanism for the spreading of information in large-scale networks where nodes periodically contact each other in a random fashion, exchanging part of their local information. They also form the basis for higher level interaction between nodes in large CAS. Besides mean field model checking procedures, FlyFast also provides two kinds of stochastic simulation procedures: one based on standard individual probabilistic simulation and one based on fast simulation [7]. The latter uses a mean-field approximation to simulate the behaviour of a single object in a large population.

2   Gossip Protocol

As a running example we consider the gossip shuffle protocol of [1,2] that we briefly recall in the following. In particular, as in [1], we analyse the dissemination of a generic data item d in a *fully connected network* in which the nodes execute the shuffling protocol. We consider the discrete time variant of this protocol with a maximal delay between two subsequent gossips of a node denoted by G_{max}. Following the mean field approximation technique [1,2,7] the behaviour of an individual node is based on its local state and the current occupancy measure vector.

Figure 1 shows the states and transitions of a single node where $G_{max} = 3$ due to space limitations. The red states, $D0$ and $O0$, denote states in which the gossip node is active, i.e. it can initiate an exchange of local information with a passive node; in $D0$ (resp. $O0$) the node has (resp. does not have) the

---

[1] More specifically, the occupancy measure vector consists of a number of elements equal to the number of local states of an object, providing, for each state, the fraction of objects in the total population that are currently in that state.

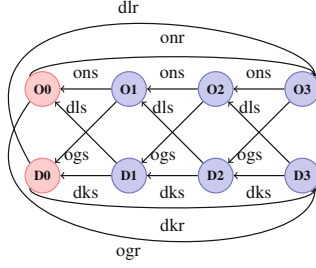[2] http://quanticol.github.io/jSAM/.

Fig. 1. Push-pull gossip model of individual gossip node with rounds of length 3 (i.e. $g_{max} = 3$). Active states are red, passive ones blue. (Color figure online)

data element in its local store. The blue states denote states in which the node is passive and can be contacted by an active node. The D/O convention w.r.t. having the data element applies also to the passive nodes. For further details of the model the reader is referred to [1,2].

3   The FlyFast Population Modelling Language

The modelling language of FlyFast consists of basic constructs to describe the probabilistic behaviour of an individual object, such as constants, states, action probabilities and transitions. The constants in the gossip model are the total number of nodes N, the number of different data elements in the system n, the size of the cache c and the number of data elements exchanged between two shuffling nodes s. Their definition is shown in Fig. 2. Furthermore, the action probabilities make use of a number of conditional probabilities, expressed in terms of the constants n, c and s. For example, P_01_10 stands for P(01|10) and denotes the conditional probability that after a shuffle the active node looses the data element, whereas the passive node acquires it (the '01' part of P_01_10) given that before the shuffle the active node had the data element and the passive one did not (the '10' part of P_01_10, see [1,2] for details).

Action probabilities are defined as shown in Fig. 3. The action labels are those of Fig. 1. For example, the action dlr ('has **d**, **l**ooses it and **r**esets gossip delay') labels the transition from the active state in which the object has the d-element ($D0$) to the passive state without d in which the clock is reset to G_{max}, i.e. $O3$

const N $= 2500$	**const** P_01_10 $= \frac{s}{c} * \frac{n-c}{n-s}$	**const** P_11_10 $=$ P_11_01
const n $= 500$	**const** P_10_01 $=$ P_01_10	**const** P_01_11 $= \frac{s}{c} * \frac{c-s}{c} * \frac{n-c}{n-s}$
const c $= 100$	**const** P_01_01 $= \frac{c-s}{c}$	**const** P_10_11 $=$ P_01_11
const s $= 50$	**const** P_10_10 $=$ P_01_01	**const** P_11_11 $= 1 - 2 * \frac{s}{c} * \frac{c-s}{c} * \frac{n-c}{n-s}$
	const P_11_01 $= \frac{s}{c} * \frac{c-s}{n-s}$	**const** P_00_00 $= 1$

Fig. 2. Constants of the FlyFast Gossip model.

action dlr :
$(\mathsf{frc}(O1) + \mathsf{frc}(O2) + \mathsf{frc}(O3)) * \mathsf{P\_01\_10} * e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))} +$
$(\mathsf{frc}(D1) + \mathsf{frc}(D2) + \mathsf{frc}(D3)) * \mathsf{P\_01\_11} * e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))})$
action dls :
$\mathsf{frc}(O0) * \mathsf{P\_10\_01} * e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))} + \mathsf{frc}(D0) * \mathsf{P\_10\_11} * e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))}$
action dkr :
$(\mathsf{frc}(O1) + \mathsf{frc}(O2) + \mathsf{frc}(O3) + \mathsf{frc}(D1) + \mathsf{frc}(D2) + \mathsf{frc}(D3)) * (1 - e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))}) +$
$(\mathsf{frc}(O0) + \mathsf{frc}(D0)) +$
$(\mathsf{frc}(O1) + \mathsf{frc}(O2) + \mathsf{frc}(O3)) * (\mathsf{P\_10\_10} + \mathsf{P\_11\_10}) * e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))} +$
$(\mathsf{frc}(D1) + \mathsf{frc}(D2) + \mathsf{frc}(D3)) * (\mathsf{P\_10\_11} + \mathsf{P\_11\_11}) * e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))}$
action dks :
$(\mathsf{frc}(O0) + \mathsf{frc}(D0)) * (1 - e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))}) + (1 - (\mathsf{frc}(O0) + \mathsf{frc}(D0))) +$
$\mathsf{frc}(O0) * (\mathsf{P\_01\_01} + \mathsf{P\_11\_01}) * e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))} +$
$\mathsf{frc}(D0) * (\mathsf{P\_01\_11} + \mathsf{P\_11\_11}) * e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))}$
action ogr : $(\mathsf{frc}(D1) + \mathsf{frc}(D2) + \mathsf{frc}(D3)) * (\mathsf{P\_10\_01} + \mathsf{P\_11\_01}) * e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))}$
action ons :
$(\mathsf{frc}(O0) + \mathsf{frc}(D0)) * (1 - e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))}) + (1 - (\mathsf{frc}(O0) + \mathsf{frc}(D0))) +$
$\mathsf{frc}(O0) * \mathsf{P\_00\_00} * e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))} + \mathsf{frc}(D0) * \mathsf{P\_10\_10} * e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))}$
action onr :
$(\mathsf{frc}(O1) + \mathsf{frc}(O2) + \mathsf{frc}(O3) + \mathsf{frc}(D1) + \mathsf{frc}(D2) + \mathsf{frc}(D3)) * (1 - e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))}) +$
$(\mathsf{frc}(O0) + \mathsf{frc}(D0)) +$
$(\mathsf{frc}(O1) + \mathsf{frc}(O2) + \mathsf{frc}(O3)) * \mathsf{P\_00\_00} * e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))} +$
$(\mathsf{frc}(D1) + \mathsf{frc}(D2) + \mathsf{frc}(D3)) * \mathsf{P\_01\_01} * e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))}$
action ogs : $\mathsf{frc}(D0) * (\mathsf{P\_01\_10} + \mathsf{P\_11\_10}) * e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))}$

Fig. 3. Actions and their probabilities in the FlyFast Gossip model

state D0 {dlr.O3 + dkr.D3}	**state** O0 {ogr.D3 + onr.O3}
state D1 {dls.O0 + dks.D0}	**state** O1 {ogs.D0 + ons.O0}
state D2 {dls.O1 + dks.D1}	**state** O2 {ogs.D1 + ons.O1}
state D3 {dls.O2 + dks.D2}	**state** O3 {ogs.D2 + ons.O2}
system mainO0 = <O0[1],O0[(N/4)-1],O1[N/4], ..., O3[N/4],D0[1]>	

Fig. 4. States and initial configuration of the FlyFast Gossip model.

in this case. The probability of action dlr depends on the occupancy measure via the quantities $\mathsf{frc}(Xi)$, with $X \in \{O, D\}$ and $i \in \{0, \ldots, 3\}$, which denote the *fraction* of objects that are in state Xi. The expression $e^{-2*(\mathsf{frc}(O0)+\mathsf{frc}(D0))}$ denotes the probability that no 'collision' occurs in the communication between two nodes, such as two active nodes that contact each other. Finally, Fig. 4 shows the definition of the states and transitions of a single node as in Fig. 1, and the non-empty elements of the initial occupancy measure vector mainO0. By default, the first element of the vector is the object selected for FlyFast analysis. We refer to [1] for further details of the model.

4 FlyFast Properties and Verification

The FlyFast syntax of bounded PCTL formulas is:

$$\Phi ::= \mathsf{ap} \mid !\Phi \mid \Phi|\Phi \mid \Phi \,\&\, \Phi \mid \mathsf{P}\{\bowtie p\}[\varphi] \qquad \text{with } \varphi ::= \mathsf{X}\,\Phi \mid \Phi\,\mathsf{U}{\le}\,k\,\Phi$$

where $\bowtie \in \{<, \le, >, \ge\}$ and ap an atomic proposition, $!,|,\&$ the usual Boolean operators, P the probabilistic path quantifier, X and U the next and until operators. These bounded PCTL formulas are interpreted over *state labelled* DTMCs in which the states consist of pairs where the first element is the local state of the selected object and the second element the limit occupancy measure vector [7]. The formal semantics can be found in [6]. FlyFast uses memoization to speed up the computation of series of path formulas where the time-bound is a parameter. For example, for a model extended in the obvious way to one in which $G_{max} = 9$, Fig. 5 shows the probability that the selected node has seen the data element within time $t \in \{0, \dots, 3000\}$:

$$\mathsf{isTrue}\ \mathsf{U} \le t\ \mathsf{hasD} \quad where \quad \mathsf{hasD} = \mathsf{inD0} \mid \ \dots \ \mid \mathsf{inD9}$$

Since all nodes have the same probabilistic behaviour, this probability corresponds to the fraction of the network that has seen the data-element within time t (i.e. the coverage and convergence). This parametric analysis required 16,997 ms on an iMAC, 2,66 GHz ICi5, with 8 GB memory (same for *any* population size $N \ge 2500$!). The results in [1] show close correspondence to those obtained[3] with FlyFast for an initial state defined as system main in Fig. 4 but for $G_{max} = 9$. Figure 6 shows an example of a parametric *nested* path formula expressing, for time-bounds $t \in \{0, \dots, 1000\}$, the probability to reach a state in which the probability to get the data element within 20 steps is greater than 0.1. The jump in the graph at $t = 700$ can be explained by the crossing of a threshold
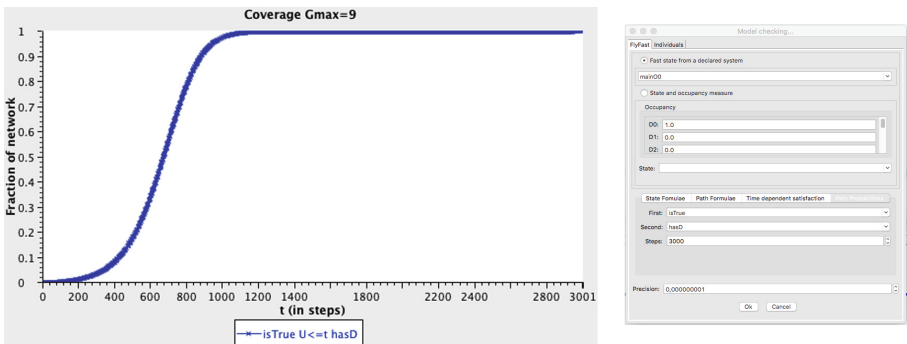

Fig. 5. Network coverage for a model with $G_{max} = 9$, $N = 2500$, $n = 500$, $s = 100$ and $c = 50$.

---

[3] Note that there is no need to extend the model with additional states that represent the fact that a node 'has seen' the data element, as was the case in [1].
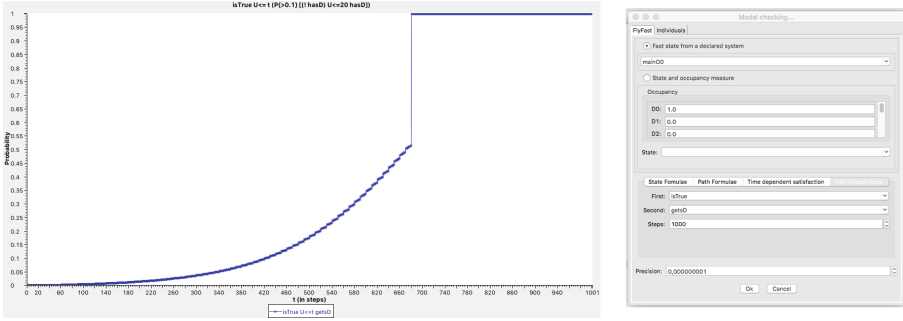
Fig. 6. Nested time-dependent probability for $G_{max} = 9$, $N = 2500$, $n = 500$, $s = 100$ and $c = 50$.

in the distribution of the data element in the network w.r.t. the bounds used in the formula[4].

$$\mathsf{isTrue\ U} \le t\ (\mathsf{P}\{> 0.1\}[!\mathsf{hasD\ U} \le 20\ \mathsf{hasD}])$$

Also *time-dependent* probabilities of (non-parametric) path formulas can be analysed. For example we may wish to make sure that in the model the probability to leave active state $O0$ in the next step is equal to 1 at any time of interest, given that such transitions model clock-ticks in this gossip model. As this probability depends on the limit occupancy measure, this may not be given for granted. However, analysis of the path formula $O0\ \mathsf{U} \le 1(D9 \mid O9)$, at different times $0, \ldots, 3000$, shows that the probability is indeed 1. Results can be visualised with the graph view in the Eclipse plugin, as in Figs. 5 and 6, or exported for customised visualisation via Gnuplot, Octave or Matlab.

5   Related Work and Conclusions

We briefly presented some features of the novel on-the-fly mean field model checker FlyFast. It scales to very large populations as the method is essentially independent of the population size (as long as it is large enough). In comparison, statistical model checking techniques scale linearly with population size. FlyFast implements a discrete time, on-the-fly probabilistic counterpart of the global fluid model checking method [3] for continuous time population models. Under some conditions, set out in [5], continuous time population models can be treated too by FlyFast applying an appropriate discretisation of the model and related CSL formulas.

References

1. Bakhshi, R.: Gossiping models - formal analysis of epidemic protocols. Ph.D. thesis, Vrije Universiteit Amsterdam, January 2011. http://www.cs.vu.nl/en/Images/Gossiping_Models_van_Rena_Bakhshi_tcm210-256906.pdf

---

[4] Model checking time: 14,720 ms.

2. Bakhshi, R., Cloth, L., Fokkink, W., Haverkort, B.R.: Mean-field framework for performance evaluation of push-pull gossip protocols. Perform. Eval. **68**(2), 157–179 (2011)
3. Bortolussi, L., Hillston, J.: Model checking single agent behaviours by fluid approximation. Inf. Comput. **242**, 183–226 (2015)
4. Latella, D., Loreti, M., Massink, M.: On-the-fly fast mean-field model-checking. In: Abadi, M., Lluch Lafuente, A. (eds.) TGC 2013. LNCS, vol. 8358, pp. 297–314. Springer, Heidelberg (2014)
5. Latella, D., Loreti, M., Massink, M.: On-the-fly fluid model checking via discrete time population models. In: Beltrán, M., Knottenbelt, W., Bradley, J. (eds.) EPEW 2015. LNCS, vol. 9272, pp. 193–207. Springer, Heidelberg (2015)
6. Latella, D., Loreti, M., Massink, M.: On-the-fly PCTL fast mean-field approximated model-checking for self-organising coordination. Sci. Comput. Program. **110**, 23–50 (2015)
7. Le Boudec, J., McDonald, D.D., Mundinger, J.: A generic mean field convergence result for systems of interacting objects. In: Fourth International Conference on the Quantitative Evaluation of Systems (QEST 2007), pp. 3–18. IEEE Computer Society (2007)