QUANTICOL

A Quantitative Approach to Management and Design of Collective and Adaptive Behaviours

quantiCOL

http://www.quanticol.eu

TR-QC-08-2014

On-the-fly Fluid Model Checking via Discrete Time Population Models

Extended Version

Revision: 1.0; September 10, 2014

Author(s): Diego Latella (CNR), Michele Loreti (IMT), Mieke Massink (CNR)

Publication date: September 10, 2014

Funding Scheme: Small or medium scale focused research project (STREP)
Topic: ICT-2011 9.10: FET-Proactive 'Fundamentals of Collective Adaptive Systems' (FOCAS)
Project number: 600708

Coordinator: Jane Hillston (UEDIN)
e-mail: Jane.Hillston@ed.ac.uk
Fax: +44 131 651 1426

Part. no.	Participant organisation name	Acronym	Country
1 (Coord.)	University of Edinburgh	UEDIN	UK
2	Consiglio Nazionale delle Ricerche – Istituto di Scienza e Tecnologie della Informazione "A. Faedo"	CNR	Italy
3	Ludwig-Maximilians-Universität München	LMU	Germany
4	Ecole Polytechnique Fédérale de Lausanne	EPFL	Switzerland
5	IMT Lucca	IMT	Italy
6	University of Southampton	SOTON	UK
7	Institut National de Recherche en Informatique et en Automatique	INRIA	France

COOPERATION

Contents

Abstract

We show that, under suitable conditions, fluid model checking bounded CSL properties of selected individuals in a continuous PEPA population model can be approximated by checking equivalent bounded PCTL formulas on corresponding objects in a discrete time, time synchronous Markov population model, using an *on-the-fly* probabilistic approach. The proposed technique is applied to a benchmark client-server case study showing promising results also for the challenging case of nested formulas with time dependent truth values. The on-the-fly results are compared to those obtained with a global fluid model checking technique.

1 Introduction

Model checking has been widely recognised as a powerful approach to the automatic verification of concurrent and distributed systems, including aspects of their performance. It consists of an efficient procedure that, given an abstract model $\mathcal{M}$ of the system, decides whether $\mathcal{M}$ satisfies a logical formula Φ, typically drawn from a temporal logic. The separation of concerns between the formal system model on one hand, and the formalisation of the property (or experiment) on the other, and the automatic derivation of the analysis facilitates the exploration of different configurations and scenarios of system behaviour. Recently, the integration of mean field and fluid approximation techniques with such formal modelling techniques [5, 17, 23], originating in theoretical computer science, has received increasing attention as a way to obtain scalable formal methods supporting the design of large scale collective adaptive systems for which performance aspects are an essential feature of their desired behaviour.

The main contribution of the present paper is to show that, under suitable assumptions[1], a stochastic model of an individual object in the context of a CTMC population model, derived from a PEPA

[1] See Theorem 5 of [6].

(Performance Evaluation Process Algebra) population model [16], satisfies a robust bounded CSL formula if and only if the probabilistic DTMC model obtained via the method described in Sect. 2.3 satisfies a bounded PCTL formula obtained from the CSL formula by scaling the time bounds according to the uniformisation rate used in the method. A further important contribution is that the DTMC model can be analysed in an *on-the-fly* manner [23], generating, from a high-level specification of the individual behaviour to verify the given formula, only as much as strictly needed of the state space.

Model checking approaches are often divided into two broad categories: *global* approaches that determine the set of all states in a model $\mathcal{M}$ that satisfy a temporal logic formula Φ, and *local* approaches in which, given a state s in $\mathcal{M}$, the procedure determines whether s satisfies Φ [8, 4]. When s is a *term of a process language*, the model checking procedure can be executed "on-the-fly", driven by the syntactical structure of s. On-the-fly algorithms are following a *top-down* approach that does not require global knowledge of the complete state space. For each state that is encountered, starting from a given state, the outgoing transitions are followed to adjacent states, constructing step by step local knowledge of the state space until it is possible to decide whether the given state satisfies the formula (or memory bounds are reached).

The global fluid model checking techniques developed in [6, 5, 17] are based on a decoupling of the behaviour of the single object (or subset of objects) from the behaviour of the global system. This leads to a time-inhomogeneous continuous time Markov Chains (ICTMC) model for such individual objects. The time-inhomogeneity leads to some complications compared to the standard global model checking algorithms for CTMC which are mainly due to the fact that the truth values of the Continuous Stochastic Logic (CSL) [1, 2] properties of such ICTMC models may depend on the time at which they are evaluated.

In this paper instead, we will not completely decouple the individual behaviour from the global system behaviour, but generate states composed of a local state of the individual object and the occupancy measure of the global system from the high-level specification of the individual in an on-the-fly manner. The resulting model can be shown to be a time-homogeneous Markov model. For continuous time stochastic models such a setting has been studied in [5]. For discrete time, time synchronous, probabilistic models this setting has been studied in [23], where the results have been used as the basis for an on-the-fly mean field model checking algorithm for discrete time probabilistic population models. The main contribution of the current paper is to relate these two results formally to develop an *on-the-fly fluid* model checking algorithm for time bounded CSL formulas and PEPA population models [16]. In particular we show that the on-the-fly approach is facilitating the calculation of the truth value of nested formulas, also in the case in which those depend on time. Experimental results obtained with FlyFast, the prototype implementation of the on-the-fly fast mean field model-checker, seem promising at least for a class of reasonably well-behaving PEPA population models.

The outline of the paper is as follows. We first introduce notation and main concepts for both continuous time and discrete time population models and their relation in Sect. 2. In Sect. 3 we briefly recall the bounded temporal logic CSL and PCTL and their formal semantics. In Sect. 4 we introduce the on-the-fly approach to fluid model checking via on-the-fly fast mean field probabilistic model checking of appropriately defined discrete time models. Sect. 5 shows model checking results for a larger benchmark client server example and compares the results with those available in the literature. Sect. 6 discusses related work and Sect. 7 presents the conclusions. Detailed proofs of the main results can be found in the Appendices A and B.

2 Population Models

We consider two types of Markov population models: continuous time models and discrete time models. In both models we assume that the population size is N and that this size remains constant during execution.

2.1 Continuous Time Population Models

For CTMC population models we adopt the notation following [5]. Let $Y_i^{(N)}(t) \in \mathcal{S}$ be the random variable representing the state of object i at time t, where $\mathcal{S} = \{1, 2, \ldots, n\}$ represents the local state space of each object. Multiple classes of agents are represented by partitioning $\mathcal{S}$ into disjoint subsets and allowing state changes only within a single class. Let $\mathbf{Q}^{(N)}(\vec{x})$ denote the $n \times n$ infinitesimal generator matrix that depends on the fraction of objects $\vec{x} \in [0,1]^n$ that are in each state of $\mathcal{S}$. The latter quantity can be computed from $Y_i^{(N)}$ as $\hat{X}_i^{(N)}(t) = \frac{1}{N}\sum_{j=1}^{N} \mathbf{1}\{Y_j^{(N)}(t) = i\}$, where $\mathbf{1}\{\ldots\}$ denotes the indicator function. $\hat{X}_i^{(N)}(t)$ is a CTMC [3] on the state space $\mathcal{D}^{(N)} = \{0, \frac{1}{N}, \frac{2}{N}, \ldots, 1\}^n$, also called the *occupancy measure*. A population model can now be defined as a structure $\hat{\mathcal{X}}^{(N)} = (\mathcal{S}, N, \mathbf{Q}^{(N)}, \vec{x}_0^{(N)})$, where $\vec{x}_0^{(N)} \in \mathcal{D}^{(N)}$ is the initial state.

The average infinitesimal variation of $\hat{\mathcal{X}}^{(N)}$, given that it is in state $\vec{x}$ is $F^{(N)}(\vec{x}) = \vec{x}^T\mathbf{Q}^{(N)}(\vec{x})$, also called the drift[2]. If, for $N \to \infty$, $\mathbf{Q}^{(N)}(\vec{x})$ converges uniformly to the Lipschitz continuous generator matrix $\mathbf{Q}(\vec{x})$, and $\vec{x}_0^{(N)} \to \vec{x}_0$, and furthermore if $\vec{x}(t)$ is the solution of the ODE $\frac{d\vec{x}}{dt} = F(\vec{x}) = \vec{x}^T\mathbf{Q}(\vec{x})$, then in the limit the two processes behave almost surely the same for a finite time horizon T [9, 20][3].

It is possible to decouple the analysis of a single object from the analysis of the global system, by letting the behaviour of the single object depend on the other objects only through the solution of the fluid ODE. This result is known as *fast simulation* [9, 25]. The stochastic behaviour of a single object can be defined as $Z^{(N)} = Y_1^{(N)}$ on state space $\mathcal{S}$, assuming we are interested in the behaviour of the first object. Note that $Z^{(N)}$ is an ICTMC. Let $z(t)$ be the ICTMC of an individual object with states in $\mathcal{S}$ such that $\Pr\{z(t+dt) = j | z(t) = i\} = q_{i,j}(\vec{x}(t))dt$ and let $\mathbf{Q}_z(\vec{x}(t)) = (q_{i,j}(\vec{x}(t)))$. We then have that for any finite horizon T and $t \leq T$ the behaviour of the single object $Z^{(N)}(t)$ tends to the behaviour of the object that senses the rest of the system only through its limit behaviour given by $\vec{x}$, i.e. $z(t)$.

Running example: Consider the simple PEPA specification of processors and resources that synchronise on a common task [16]:

$$
\begin{array}{lcl}
Proc0 & := & (task1, r_1).Proc1 \\
Proc1 & := & (task2, r_2).Proc0 \\
Res0 & := & (task1, r_1).Res1 \\
Res1 & := & (reset, s).Res0 \\
Proc0[N_p] & \underset{task1}{\bowtie} & Res0[N_q]
\end{array}
$$

where $Proc0[N_p]$ is a shorthand notation for N_p instances of process $Proc0$ in parallel, and $Res0[N_q]$ denoting N_q instances of process $Res0$ in parallel. Such population oriented PEPA specifications have been given a formal semantics based on ordinary differential equations (ODE) by Hillston in [16] and by Tribastone et al. in [27]. In particular, for a population size going to infinite, and considering only the fractions of the number of processes in their various local states (i.e. their limit occupancy measure), the ODE associated to the example specification can be given as:

$$
\begin{aligned}
\tfrac{d\,proc0(t)}{dt} &= -r_1.min(proc0(t), res0(t)) + r_2.proc1(t) \\
\tfrac{d\,proc1(t)}{dt} &= -r_2.proc1(t) + r_1.min(proc0(t), res0(t)) \\
\tfrac{d\,res0(t)}{dt} &= -r_1.min(proc0(t), res_0(t)) + s.res1(t) \\
\tfrac{d\,res1(t)}{dt} &= -s.res_1(t) + r_1.min(proc0(t), res0(t))
\end{aligned}
$$

where $proc0(t), proc1(t), res0(t)$ and $res1(t)$ denote the limit occupancy measure at time t for each local state respectively. The function min denotes the minimum function and originates from the

[2] $\vec{x}^T$ denotes the transpose of vector $\vec{x}$.

[3] The conditions on uniform convergence and Lipschitz continuity automatically hold for PEPA population models because in that case the rate functions are all piecewise linear [27].

specific definition of action synchronisation of the semantics of PEPA [16]. For example, for initial values for the fractions of processes and resources in their respective initial states $n_p = 0.5$ and $n_q = 0.5$ and the following values for the individual transition rates $r_1 = 10, r_2 = 3$ and $s = 7$, the limit occupancy measure as solution of the ODE is shown in Fig. 1(b).

The infinitesimal **Q**-matrix of an individual object that depends on the behaviour of the global system via its limit occupancy measure can be retrieved as follows (see [5]). From the PEPA semantics of the synchronisation (cooperation) operator we know that the total rate of a shared $task1$ action is given by $min(r_1.proc_0(t), r_1.res_0(t))$. The rate of an *individual process* performing a $task1$ action is then this global rate divided by the fraction of objects present in the system at time t, i.e. $proc_0(t)$. The rate of an individual process performing a $task_2$ action is simply r_2 because this action does not depend on the limit occupancy measure $\vec{x}$, where $\vec{x}^T(t) = (proc_0(t), proc_1(t), res_0(t), res_1(t))$. Similar reasoning applies to the rates of a resource object. So, in summary, we obtain the following rate functions for the **Q**-matrix of an *individual* object of type 'process' which depends on $\vec{x}(t)$:

$$\begin{aligned} \mathbf{Q}_{proc0,proc1}(\vec{x}(t)) &= r_1.min(proc0(t), res0(t))/proc0(t) \\ \mathbf{Q}_{proc1,proc0}(\vec{x}(t)) &= r_2 \end{aligned}$$

and for an *individual* object of type 'resource':

$$\begin{aligned} \mathbf{Q}_{res0,res1}(\vec{x}(t)) &= r_1.min(proc0(t), res0(t))/res0(t) \\ \mathbf{Q}_{res1,res0}(\vec{x}(t)) &= s \end{aligned}$$

The rate functions used in the Q-matrix are all continuous and bounded, at least as long as we do not divide by zero.

2.2 Discrete Time Population Models

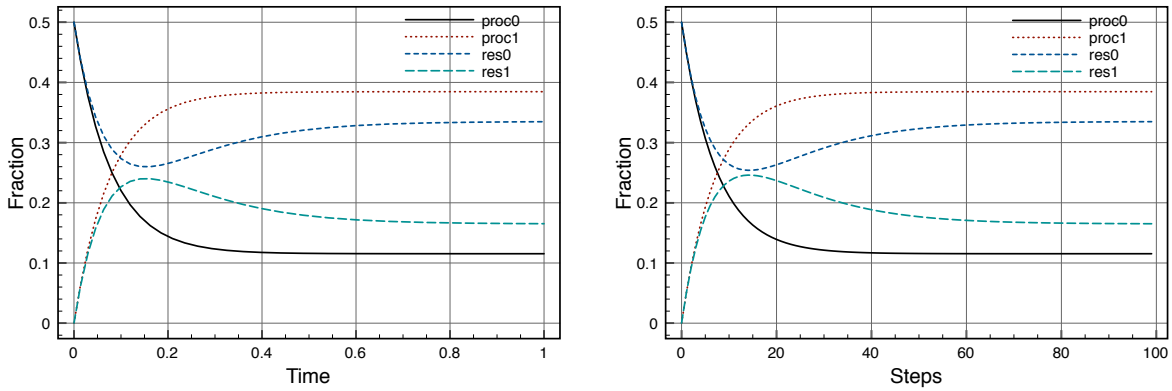
For DTMC population models we consider again a system of N interacting objects. Let $W_i^{(N)}(k) \in \mathcal{S}$ the random variable representing the state of object i at step k, where $\mathcal{S} = \{1, 2, \ldots, n\}$ represents the local state space of each object[4]. Let $\mathbf{K}^{(N)}(\vec{m})$ denote the $n \times n$ one step transition probability matrix that depends on the fraction of objects $\vec{m} \in [0,1]^n$ that are in each state of $\mathcal{S}$. This fraction can be computed as $\hat{M}_i^{(N)}(k) = \frac{1}{N}\sum_{j=1}^{N} \mathbf{1}\{W_j^{(N)}(k) = i\}$. $\hat{M}_i^{(N)}(k)$ is a DTMC [25] on the state space $\mathcal{D}^{(N)}$. A population DTMC model can now be defined as a structure $\hat{\mathcal{M}}^{(N)} = (\mathcal{S}, N, \mathbf{K}^{(N)}, \vec{m}_0^{(N)})$, where $\vec{m}_0^{(N)} \in \mathcal{D}^{(N)}$ is the initial state. The average variation of $\hat{\mathcal{M}}^{(N)}$, given that it is in state $\vec{m}$ is $F^{(N)}(\vec{m}) = \vec{m}^T\mathbf{K}^{(N)}(\vec{m})$. If, for all i, j and for $N \to \infty$, the elements $\mathbf{K}_{i,j}^{(N)}(\vec{m})$ converge uniformly in $\vec{m}$ to some $\mathbf{K}_{i,j}(\vec{m})$, which is a continuous function of $\vec{m}$, and $\vec{m}_0^{(N)}$ converges almost surely to $\vec{m}_0$, and furthermore define $\vec{m}(k)$ as follows: $\vec{m}(0) = \vec{m}_0$ and $\vec{m}(k+1) = \vec{m}(k)^T\mathbf{K}(\vec{m}(k))$; then, for any fixed step t, almost surely, the two processes behave the same [25]. As for CTMC population models, it is possible to decouple the analysis of the single object from the analysis of the global system using a fast simulation approach involving the solution of a difference equation rather than an ODE.

Example: Taking probabilities α_i for the rates r_i in the processes and resources example, we obtain the following difference equations for $\vec{m}^T(k) = (m_{p0}(k), m_{p1}(k), m_{r0}(k), m_{r1}(k)))$:

$$\begin{aligned} m_{p0}(k+1) &= m_{p0}(k) - \alpha_1.min(m_{p0}(k), m_{r0}(k)) + \alpha_2.m_{p1}(k) \\ m_{p1}(k+1) &= m_{p1}(k) + \alpha_1.min(m_{p0}(k), m_{r0}(k)) - \alpha_2.m_{p1}(k) \\ m_{r0}(k+1) &= m_{r0}(k) - \alpha_1.min(m_{p0}(k), m_{r0}(k)) + \alpha_s.m_{r1}(k) \\ m_{r1}(k+1) &= m_{r1}(k) + \alpha_1.min(m_{p0}(k), m_{r0}(k)) - \alpha_s.m_{r1}(k) \end{aligned}$$

where $m_{pj}(k)$ and $m_{rj}(k)$ denote the limit occupancy measure at step k for processes and resources. We can also retrieve the one step probability matrix for each individual process and resource object

[4] As for CTMC population models, multiple classes of agents are represented by partitioning S into disjoint subsets and allowing state changes only within a single class.



(a) Solution of the ODE for $r_1 = 10, r_2 = 3$, $s = 7$, $n_p = 0.5$ and $n_q = 0.5$

(b) Solution of the difference equations for $\alpha_1 = 0.1, \alpha_2 = 0.03$, $\alpha_s = 0.07$, $n_p = 0.5$ and $n_q = 0.5$

Figure 1: Occupancy measure of process and resource states

using a similar reasoning as in the CTMC case:

$$\begin{aligned} \mathbf{K}_{p0,p1}(\vec{m}(k)) &= \alpha_1.min(m_{p0}(k), m_{r0}(k))/m_{p0}(k) \\ \mathbf{K}_{p1,p0}(\vec{m}(k)) &= \alpha_2 \\ \\ \mathbf{K}_{r0,r1}(\vec{m}(k)) &= \alpha_1.min(m_{p0}(k), m_{r0}(k))/m_{r0}(k) \\ \mathbf{K}_{r1,r0}(\vec{m}(k)) &= \alpha_s \end{aligned}$$

Note that the difference equations can be obtained from $\mathbf{K}$ by $\vec{m}(k+1) = \vec{m}(k)^T.\mathbf{K}(\vec{m}(k))$.

2.3 Relation Between the Models

It is possible to establish a close behavioural relation between the continuous and the discrete time population models by deriving an appropriate value for the probabilities α from the rates in the continuous model. This derivation takes two aspects into consideration. First of all, we want that the discrete model of the individual objects has the same local states as the individuals in the continuous model. A well-known approach to obtain this is by uniformisation. The second observation is that the difference equations should give a solution that provides an acceptable approximation of the solution of the original ODEs. This is possible by recognising that we can interpret the difference equations as an instance of the Euler forward method for solving ODEs under the condition that a suitable step size can be found that guarantees absolute stability of the method and a sufficient accuracy (see for example [26]).

Uniformisation involves finding a uniformisation rate q that is *at least as large* as the maximal exit rate of the states in the CTMC at hand, obtaining the one step probability matrix $\mathbf{K}$ of a DTMC by $\mathbf{K} = \mathbf{I} + \frac{1}{q} \cdot \mathbf{Q}$, where $\mathbf{Q}$ is the infinitesimal rate matrix. Note that in our case the rates in $\mathbf{Q}$ may depend on the occupancy measure $\vec{m}$. However, $0 \leq m_i \leq 1$ for all $i \in |\mathcal{S}|$, so assuming rate-functions that include minimum functions and linear combination (but not rational functions)[5] that derive from PEPA specifications we can easily find a suitable q given that the occupancy measure is at most 1. The $\mathbf{Q}$ matrix we intend here includes the rates of the possibly multiple classes of objects.

The uniformisation rate might, however, not be large enough to lead to stable and accurate calculations in the difference equations that are derived from the one step transition matrix of the individual objects obtained by uniformisation. First we have to check that the global error does not grow exponentially. For linear systems of l differential equations where $u(t) \in \mathbb{R}^l$ and $d\ u(t)/dt = Au$ where A

[5]These are piecewise linear functions leading to the class of split-free PEPA models [14].

$$
\begin{array}{ll}
s,t \models_{\mathcal{M}} a & \Leftrightarrow a \in \ell(s) \\
s,t \models_{\mathcal{M}} \neg\Phi & \Leftrightarrow \text{not } s,t \models_{\mathcal{M}} \Phi \\
s,t \models_{\mathcal{M}} \Phi_1 \vee \Phi_2 & \Leftrightarrow s,t \models_{\mathcal{M}} \Phi_1 \text{ or } s,t \models_{\mathcal{M}} \Phi_2 \\
s,t \models_{\mathcal{M}} \mathcal{P}_{\bowtie p}(\varphi) & \Leftrightarrow \Pr\{\sigma \in \mathit{Paths}_{\mathcal{M}}(s,t) \mid \sigma,t \models_{\mathcal{M}} \varphi\} \bowtie p \\
\sigma,t \models_{\mathcal{M}} \Phi_1 \mathcal{U}^{\leq\tau} \Phi_2 & \Leftrightarrow \exists\, \tau_2 \text{ s.t. } 0 \leq \tau_2 \leq \tau, \\
 & \quad \sigma@\tau_2, t+\tau_2 \models_{\mathcal{M}} \Phi_2 \ \wedge \\
 & \quad \forall\, 0 \leq \tau_1 < \tau_2,\ \sigma@\tau_1, t+\tau_1 \models_{\mathcal{M}} \Phi_1
\end{array}
$$

Table 1: Satisfaction relation for the fragment of CSL.

is an $l \times l$ matrix a necessary condition is that $h\lambda$ is in the stability region of the Euler method for each eigenvalue λ of matrix A and step size h. So, for each eigenvalue λ we need that $|1 + h\lambda| \leq 1$, which implies that $-2 \leq h\lambda \leq 0$ [26]. For non-linear systems we need to determine the range of each eigenvalue and make sure that the step size h is taken small enough so that $h\lambda$ stays within the region of absolute stability for its complete range.

Example: For the running example, with $r_1 = 10, r_2 = 3.0$ and $s = 7.0$, we obtain uniformisation rate $q = 10$ and eigenvalues $\lambda_1 = 0$ or $-r_1 - r_2 \leq \lambda_1 \leq -r_2$ and $\lambda_2 = 0$ or $-r1 - s \leq \lambda_2 \leq -s$, showing that all eigenvalues are in a bounded range, with a maximum absolute value of 17. So when taking $h = 1/q$ we get that $0 \leq 17 * 1/q \leq 2$. This implies that $q = 10$ guarantees stability of the method. However, to obtain better accuracy we may choose a higher value, for example $q = 100$. Also empirically this gives a close correspondence of the solution of the difference equation to that of stochastic simulation of the same model for large N. For example, for initial values for the fractions of processes and resources in their respective initial states $n_p = 0.5$ and $n_q = 0.5$ and the following values for the individual transition rates $r_1 = 10, r_2 = 3$ and $s = 7$ as before, the solution of the difference equations, for $\alpha_1 = 10/q, \alpha_2 = 3/q$ and $\alpha_s = 7/q$ is shown in Fig. 1(b).

Our aim is now to use the individual *discrete time* model to verify bounded temporal logic properties of an individual in the *continuous model* via an on-the-fly fast mean field model checking procedure for discrete time synchronous population models presented in [23].

3 Individual Objects Properties

Properties of the behaviour of individuals in the context of a large population model can be expressed as formulas of a suitable temporal logic. For the purpose of this paper, properties of continuous models are expressed in bounded CSL (Continuous Stochastic Logic) [1, 2], and properties of discrete models are expressed in bounded PCTL (Probabilistic Computation Tree Logic) [13].

3.1 Continuous Stochastic Logic for ICTMC

Given a set $\mathscr{P}$ of atomic propositions, the syntax of the fragment of bounded CSL we consider is defined below, where $a \in \mathscr{P}$, $\tau \in \mathbb{Q}_{\geq 0}$ and $\bowtie \in \{>, <\}$ and $p \in [0,1] \cap \mathbb{Q}$:

$$\Phi ::= a \mid \neg\,\Phi \mid \Phi \ \vee\ \Phi \mid \mathcal{P}_{\bowtie p}(\varphi) \text{ where } \varphi ::= \Phi\,\mathcal{U}^{\leq\tau}\,\Phi.$$

CSL formulas are interpreted over state labelled (I)CTMCs $\langle \mathcal{M}, \ell \rangle$, where $\mathcal{M}$ is an (I)CTMC with state set $\mathcal{S}$ and $\ell : \mathcal{S} \to 2^{\mathscr{P}}$ associates each state with a set of atomic propositions. We define the satisfaction relation on $\mathcal{M}$ and the logic in Table 1. We abbreviate $\langle \mathcal{M}, \ell \rangle$ with $\mathcal{M}$, when no confusion can arise, with $\mathbf{Q}$ its infinitesimal generator matrix. A path σ over $\mathcal{M}$ is a non-empty sequence of states $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \cdots$ such that the probability of going from s_i to s_{i+1} at time $T_i = \sum_{j=0}^{i} t_i > 0$ for all $i \geq 0$. We let $\mathit{Paths}_{\mathcal{M}}(s,t)$ denote the set of all infinite paths over $\mathcal{M}$ starting from state s at time t. We require that all subsets of paths considered are measurable. By $\sigma[i]$ we denote the element s_i

$s \models_{\mathcal{M}} a$	$\Leftrightarrow$	$a \in \ell(s)$
$s \models_{\mathcal{M}} \neg\Phi$	$\Leftrightarrow$	not $s \models_{\mathcal{M}} \Phi$
$s \models_{\mathcal{M}} \Phi_1 \vee \Phi_2$	$\Leftrightarrow$	$s \models_{\mathcal{M}} \Phi_1$ or $s \models_{\mathcal{M}} \Phi_2$
$s \models_{\mathcal{M}} \mathcal{P}_{\bowtie p}(\varphi)$	$\Leftrightarrow$	$\Pr\{\sigma \in Paths_{\mathcal{M}}(s) \mid \sigma \models_{\mathcal{M}} \varphi\} \bowtie p$
$\sigma \models_{\mathcal{M}} \Phi_1 \mathcal{U}^{\leq k} \Phi_2$	$\Leftrightarrow$	$\exists\, 0 \leq h \leq k \; s.t. \; \sigma[h] \models_{\mathcal{M}} \Phi_2 \; \wedge$ $\forall\, 0 \leq i < h \; . \; \sigma[i] \models_{\mathcal{M}} \Phi_1$

Table 2: Satisfaction relation for the fragment of PCTL.

of path σ. Finally, in the sequel we will consider ICTMCs equipped with an initial state s_0, i.e. the probability mass is initially all in s_0.

Example: An example of a CSL property for the processes and resources model is the state formula $\mathcal{P}_{>p}(Proc0\, \mathcal{U}^{\leq t}\, Proc1)$ expressing that a process will manage to get a resource within t time units with probability at least p.

3.2 Probabilistic Logic for DTMC

Given a set $\mathscr{P}$ of atomic propositions, the syntax of the fragment of bounded PCTL we consider is defined below, where $a \in \mathscr{P}$, $k \in \mathbb{N}$ and $\bowtie \in \{>, <\}$ and $p \in [0,1] \cap \mathbb{Q}$:

$$\Phi ::= a \mid \neg\,\Phi \mid \Phi \;\vee\; \Phi \mid \mathcal{P}_{\bowtie p}(\varphi) \text{ where } \varphi ::= \Phi\, \mathcal{U}^{\leq k}\, \Phi.$$

PCTL formulas are interpreted over *state labelled* DTMCs in a similar way as for CTMCs. We assume $\mathbf{P}$ to be the one step probability matrix for $\mathcal{M}$. A path σ over $\mathcal{M}$ is a non-empty sequence of states $s_0, s_1, \cdots$ where $\mathbf{P}_{s_i,s_{i+1}} > 0$ for all $i \geq 0$. We let $Paths_{\mathcal{M}}(s)$ denote the set of all infinite paths over $\mathcal{M}$ starting from state s. Will consider DTMCs equipped with an initial state s_0. We define the satisfaction relation on $\mathcal{M}$ and the logic in Table 2.

Example: An example of a PCTL property for the processes and resources model similar to that for CSL is the state formula $\mathcal{P}_{>p}(Proc0\, \mathcal{U}^{\leq k}\, Proc1)$ expressing that a process will manage to get a resource within k *steps* (instead of time units), with probability at least p.

4 On-the-fly Fluid Model Checking via Discrete Time Models

On-the-fly fast mean field model checking is a model checking technique that exploits deterministic approximation and fast simulation results, but in the context of *discrete time, time synchronous probabilistic population models* such as those introduced in [25].

The main result of the current paper is to show that, under the assumptions of Theorem 5 of [6], a stochastic model derived from a PEPA population model satisfies a robust bounded CSL formula if and only if the appropriately uniformised probabilistic population model satisfies a bounded PCTL formula obtained from the CSL formula by scaling the time bounds according to the uniformisation rate.

We first define two transformation functions. Function $\mathcal{T}_M$ takes an ICTMC $z(t)$ with infinitesimal generator matrix $\mathbf{Q}(t)$ and initial state s_0. It takes a step size $d \in \mathbb{Q}$ and a time bound $b > d$. It returns a DTMC with state set $\mathcal{S} \times \{0, \ldots, \lfloor \frac{b}{d} \rfloor\}$, initial state $(s_0, 0)$ and one step transition probability matrix $\mathbf{U}$ as follows:

Definition 1. *For all $0 < d \in \mathbb{Q}, b \in \mathbb{R}$ with $b > d$, and infinitesimal generator matrix $\mathbf{Q}(t)$, $\mathcal{T}_M(\mathbf{Q}(t), d, b)$ is the one step transition probability matrix $\mathbf{U}$, defined by:*

$$\mathbf{U}_{(s,i),(s',i')} = \begin{cases} [\mathbf{I} + d{\cdot}\mathbf{Q}(i{\cdot}d)]_{s,s'}, \textit{if } i' = i{+}1, \mathbf{Q}(i{\cdot}d)_{s,s} \neq 0, \\ 1, \textit{if } i' = i, s' = s, \mathbf{Q}(i{\cdot}d)_{s,s} = 0, \\ 0, \textit{otherwise} \end{cases}$$

where the indexes of $\mathbf{U}$ *are assumed to be ordered as follows:*

$$(s_0,0),...,(s_n,0),(s_0,1),...,(s_n,1),...,(s_0,\lfloor\frac{b}{d}\rfloor),...,(s_n,\lfloor\frac{b}{d}\rfloor).$$

Function $\mathcal{T}_F$ transforms bounded CSL into bounded PCTL formulas.

Definition 2. *For atomic propositions* a, in_s, *bounded CSL formulas* Φ, Φ_1 *and* Φ_2, *and* $d \in \mathbb{Q}$, *function* $\mathcal{T}_F$ *is defined as follows:*

$$\begin{aligned}
&\mathcal{T}_F[\![a]\!]_d = a\\
&\mathcal{T}_F[\![\neg\Phi]\!]_d = \neg\mathcal{T}_F[\![\Phi]\!]_d\\
&\mathcal{T}_F[\![\Phi_1 \vee \Phi_2]\!]_d = \mathcal{T}_F[\![\Phi_1]\!]_d \vee \mathcal{T}_F[\![\Phi_2]\!]_d\\
&\mathcal{T}_F[\![\mathcal{P}_{\bowtie p}(\Phi_1\ \mathcal{U}^{\leq\tau}\ \Phi_2)]\!]_d = \mathcal{P}_{\bowtie p}(\mathcal{T}_F[\![\Phi_1]\!]_d\ \mathcal{U}^{\leq\lfloor\frac{\tau}{d}\rfloor}\ \mathcal{T}_F[\![\Phi_2]\!]_d)
\end{aligned}$$

where atomic proposition in_s *is defined as* $(s',w) \models_{\mathcal{M}} in_s$ *iff* $s' = s$ *for any DTMC* $\mathcal{M}$ *with states in* $\mathcal{S} \times W$, *for some set* W.

Bounded CSL until formulas translate to bounded PCTL until formulas with the same probability bound and structure, but with a time bound $\frac{\tau}{d}$ where τ was the original time bound in the CSL formula. In the sequel, we let $|\Phi|$ denote the *duration* of Φ, i.e. the length of time to which it refers, as follows:

Definition 3. *For any bounded CSL formula* Φ *the duration of* Φ, $|\Phi|$ *is defined as follows:*

$$\begin{aligned}
&|a| = 0\\
&|\neg\Phi| = |\Phi|\\
&|\Phi_1 \vee \Phi_2| = \max\{|\Phi_1|,|\Phi_2|\}\\
&|\mathcal{P}_{\bowtie p}(\Phi_1\ \mathcal{U}^{\leq\tau}\ \Phi_2)| = \tau + \max\{|\Phi_1|,|\Phi_2|\}
\end{aligned}$$

Recall that we assume that time bounds in until formulas are *rational* numbers. For formula Φ, we let $\vec{\tau}_\Phi = (\tau_1,\ldots,\tau_l)$, denote the vector of all time bounds occurring in the (until subformulae of) Φ; furthermore we define d_Φ and D_Φ as follows: $d_\Phi = \max\{d \in \mathbb{Q} \mid \frac{\tau_j}{d} \in \mathbb{N}, \text{ for } j = 1\ldots l\}$ and $D_\Phi = \{d \in \mathbb{Q} \mid \text{ there exists } w \in \mathbb{N} \text{ s.t. } d = \frac{d_\Phi}{w}\}$. Note that d_Φ is well defined since $\tau_j \in \mathbb{Q}$, for $j = 1\ldots l$; actually, letting $\tau_i = \frac{a_i}{b_i}$, such that $\gcd(a_i,b_i) = 1$, it is easy to see that $d_\Phi = \frac{1}{\mathrm{lcm}(b_1,\ldots,b_l)}$ where MCM denotes the maximal common multiple. We are now ready to state the main Theorem for robust CSL formulas[6]:

Theorem 1. *Let* $\mathcal{X}^{(N)}$ *be a sequence of CTMC population models, with deterministic fluid limit* $\vec{x}(t)$ *for any fixed time* $t < T$, *under initial condition* $\vec{x}(0) = \vec{x}_0$, *and let* $z = z(t)$ *be the stochastic process defined from* $\mathcal{X}^{(N)}$ *as in Sect. 2.1. Let* Φ *be a robust CSL formula for* z. *There exists* $N_0 \in \mathbb{N}$, *such that, for all* $d \in D_\Phi$, *with* $d \leq \frac{1}{q}$ *as in Sect. 2.3 and for all* $N \geq N_0$ *and* $b > \lceil\frac{|\Phi|}{d}\rceil$ *the following holds:*

$$s,t \models_z \Phi \text{ iff } (s,\lfloor\frac{t}{d}\rfloor) \models_{\mathcal{T}_M(z,d,b)} \mathcal{T}_F[\![\Phi]\!]_d$$

The proof is by induction on the structure of the bounded CSL formula Φ. One is usually interested in the result for $t = 0$. A detailed proof is provided in Annex B.

The result of Theorem 1 shows that it is indeed possible, under suitable conditions, to use PCTL and a discrete time Markov population model to obtain similar results as by global fluid model checking CSL formulas on ICTMCs. There are a few reasons why this is useful. First of all, the ICTMC model can be translated into an *individual IDTMC model*, from which the full DTMC $\mathbf{U}$ can be generated in an on-the-fly manner, generating only those states that are required for the verification of the property of interest. Second, we can reuse the probabilistic fast mean-field model-checker FlyFast for

[6] We refer to [6] for the definition of formula robustness and to [20, 7] for constraints on time horizon T.

which a prototype implementation exists (see [23]) that, in turn, is based on an efficient on-the-fly probabilistic model checker [24]. Finally, the approach outlined in this paper does not require the *a priori* calculation of discontinuity points, i.e. points in time in which the truth values of time-dependent (sub)-formulas of the until formula change [5]. Such points are detected on-the-fly during calculation of the probabilities, up to a difference of the size of the step size.

The proof of Theorem 1 also provides insight in where the various approximation steps are applied, and therefore in where approximation errors may get introduced or where the method could be further improved.

- If discontinuity time points were available *a priori*, one could do with much fewer model checking invocations, since in between the discontinuity points the sets of goal states and unsafe states do not change. This is the approach followed in the global fluid model checking approach in [5]. The on-the-fly model checking method gives an approximation also when these discontinuity times are not available and uses a model checking procedure reusing model checking results of earlier checks in an efficient way. Which of the two ways is more efficient depends also on the computational effort needed to compute the discontinuity points. The latter involves finding all zeros of an analytic function.

- We need a balanced estimate of a suitable step size. If the step size is much smaller than required for uniformisation and an acceptable approximation of the ODE solution, then the model checker is invoked more often than necessary. This is particularly so for nested formulas. Moreover, model checking times will also increase considerably when large time bounds are used in the CSL formulas since they will be transformed into PCTL formulas with a high number of steps in the time bound.

- The quality of the results depend also on whether the Euler method is absolutely stable [26] for the particular model, property and chosen step size, as well as whether the local error in each step is sufficiently small. This requires a careful analysis of the model before applying the model checking technique and an awareness from those using the model checker of the specific class of models for which the technique gives reliable results. Suitable guidance provided by an integrated tool suite could be very useful in this case.

- An open issue is whether instead of the first order Euler method higher order approximation methods or adaptive step size methods could be used in combination with the on-the-fly model checking technique, and what their effect would be on the computational efficiency and accuracy of the results.

Let us continue with some model checking results for the running example and a comparison of a more extended benchmark client-server example in the following sections.

4.1 Using FlyFast to Approximate Fluid Model Checking Results

The on-the-fly probabilistic model checking algorithm implemented in FlyFast abstracts from any specific language and different semantic interpretations of a language. Only an abstract interpreter function is assumed that, given a generic process term, returns a probability distribution over the set of terms. Below, we let proc be the (generic) type of *probabilistic process terms* while we let formula and path_formula be the types of *state-* and *path-* PCTL formulas. Finally, we use lab to denote the type of *atomic propositions*.

The abstract interpreter can be modelled by means of two functions: next and lab_eval. Function next associates a list of pairs $(\mathsf{proc}, \mathsf{float})$ to each element of type proc. The list of pairs gives the terms, i.e. states, that can be reached in one step from the given state and their one-step transition probability. We require that for each s of type proc it holds that $0 < p' \leq 1$, for all $(s', p') \in \mathsf{next}(s)$ and

```
action task1p  :  min(α_task1 * frc(Proc0), α_task1 * frc(Res0))/frc(Proc0);
action task1r  :  min(α_task1 * frc(Proc0), α_task1 * frc(Res0))/frc(Res0);
action task2   :  α_task2;
action reset   :  α_reset;

state Proc0{task1p.Proc1}                    state Res0{task1r.Res1}
state Proc1{task2.Proc0}                     state Res1{reset.Res0}

system mySystem =< Proc0[1000], Res0[1000] >
```

Table 3: Processes and Resources specification in FlyFast with uniformisation rate $q = 100$, and the following values for the probabilities (not shown in the Table): $\alpha_{\text{task1}} = 10.0/q$, $\alpha_{\text{task2}} = 3.0/q$ and $\alpha_{\text{reset}} = 7.0/q$.

$\sum_{(s',p')\in \mathsf{next}(s)} p' = 1$. Function lab_eval returns for each element of type proc a function associating a bool to each atomic proposition a in lab. Each instantiation of the algorithm consists in the appropriate definition of next and lab_eval, depending on the language at hand and its semantics.

The local model-checking algorithm is defined as a function, Check. On atomic state-formulas, the function returns the value of lab; when given a non-atomic state-formula, Check calls itself recursively on sub-formulas, in case they are state-formulas, whereas it calls function CheckPath, in case the sub-formula is a path-formula. In both cases the result is a Boolean value that indicates whether the state satisfies the formula.

Function CheckPath takes a state $s \in$ proc and a PCTL path-formula $\varphi \in$ path_formula as input. As a result, it produces the probability measure of the set of paths, starting in state s, which satisfy path-formula φ.

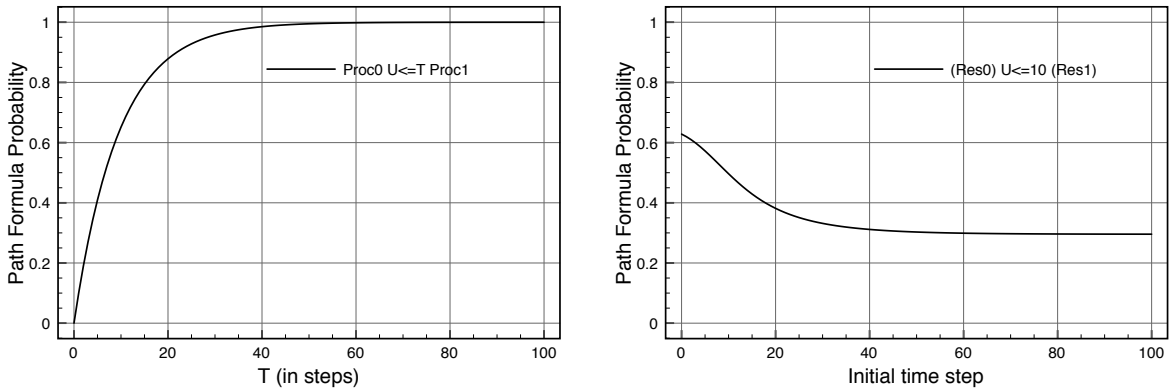
On-the-fly *fast* PCTL approximated model-checking on the limit DTMC $\mathbf{U}$ is obtained by instantiating proc with $\mathcal{S} \times \mathcal{U}^{|\mathcal{S}|}$ where $\mathcal{U}^{|\mathcal{S}|}$ is the set of all occupancy measures over the local state set $\mathcal{S}$, and lab with the set of atomic propositions $\mathscr{P}$; next is instantiated with $\mathsf{next}_{\mathbf{U}}$, for $C, C' \in \mathcal{S}$, defined as follows:

$$\mathsf{next}_{\mathbf{U}}(\langle C, \vec{m}\rangle) = [(\langle C', \vec{m} \cdot \mathbf{K}(\vec{m})\rangle, p') \mid \mathbf{K}(\vec{m})_{C,C'} = p' > 0],$$

with $\mathbf{K}(\vec{m})$ as in Theorem 4.1 of [25] and in Sect. 2.2, i.e. the one step probability matrix of an IDTMC in which the probabilities may depend on the limit occupancy measure; lab_eval is instantiated as expected. Note that, in the worst case, $\mathsf{next}_{\mathbf{U}}(\langle C, \mathbf{m}\rangle)$ returns $|\mathcal{S}|$ states only, due to the collapse of the occupancy measure vectors to a single one, at the relevant step. For further details we refer to [23].

Going back to our running example of processors and resources, in Table 3 we show the model as defined in the FlyFast input language. Note that in the probability functions associated to the action in the table one has to make sure that no division by zero ever occurs. In our running example, this is indeed the case, as can be seen from the transient occupancy measure values of *Proc0* and *Res0* shown in Fig. 1. The constants α_x in Table 3 denote the probability of the activities x obtained from uniformisation of the rates involved in the PEPA model, $\mathsf{frc}(X)$ denotes the fraction of the population in state X and $< Proc0[1000], Res0[1000] >$ denotes the initial system state composed of 1000 processors and 1000 resources in their respective initial states. The definitions state $X_1\{action.X_2\}$ define the possible state transitions of the individual objects. We recall that the uniformisation rate is $q = 100$ (see Sect. 2.3). Suppose that we would be interested in the verification of the CSL path formula $\phi =$ Proc0 $\mathcal{U}^{\leq t}$ Proc1 for a selected processor object in local state Proc0 of the example PEPA population model. Using Def. 2 we can translate the CSL formula into an equivalent, properly scaled PCTL path formula $\phi' =$ Proc0 $\mathcal{U}^{\leq \lfloor t \cdot q \rfloor}$ Proc1 so that we can approximate the probability of ϕ by evaluating ϕ' on the discrete time model specified (see Table 3) in the input language of the prototype FlyFast fast mean-field PCTL model-checker [23].

Fig. 2(a) shows the probability mass of the PCTL path formula Proc0 $\mathcal{U}^{\leq T}$Proc1 for values of $T = \lfloor t \cdot q \rfloor$ ranging from 0 to 100 corresponding to the CSL formula Proc0 $\mathcal{U}^{\leq t}$ Proc1 with t ranging



(a) Time dependent analysis of Proc0 $\mathcal{U}^{\leq T}$Proc1 for time bound upto 100 steps

(b) Analysis of Res0 $\mathcal{U}^{\leq 10}$ Res1 for initial times from 0 to 100

Figure 2: FlyFast model checking results for processors and resources model

from 0 to 1.

Example: Fig. 2(b) shows how the probability mass of the PCTL path formula Res0 $\mathcal{U}^{\leq 10}$ Res1 changes when it is evaluated at different initial times of the global system within 10 steps. This formula shows the probability that a specific selected resource is assigned to a process. The formula corresponds to the CSL formula Res0 $\mathcal{U}^{\leq 0.1}$ Res1 for initial times ranging from 0 to 1. Note that the satisfaction probability of the path formula changes depending on the initial time (step) at which it is evaluated. This also implies that a state formula containing such a path formula as sub-formula, for example $\mathcal{P}_{<0.4}$(Res0 $\mathcal{U}^{\leq 10}$ Res1), changes its truth value, in this case around time step 16, from false to true. This is exactly what we would expect and in line with similar observations using Fluid Model Checking of time-inhomogeneous CTMC models [6].

In Sect. 5 we provide a comparison for a larger Client-Server example of our results and those obtained with Fluid Model Checking as developed by Bortolussi and Hillston in [6].

5 Example and Comparison

Let us now consider the analysis of a larger example which is the client-server model presented in the paper introducing fluid model checking by Bortolussi and Hillston [6, 5]. The PEPA model consists of two processes, each composed of four states. One denotes a Client (see Fig. 3) which in the initial state (CQ) can only perform a request (rq) to the server and then waits (CW) for either a timeout (to) or a reply (rp) from the server to happen. After a timeout it goes to a state to recover (CR), and then returns to the initial state when recovery completed (rc). After receiving a reply (rp) the Client enters a thinking state (CT) after which it returns to the initial state upon completing thinking (th). The Server process (see Fig. 3) is initially (SQ) ready to receive a request (rq) from a Client. If it receives it, either a timeout (top) may occur or it may process the request (pr) moving to the reply state (SR). From the latter it may either produce a timeout (tor) or deliver a reply (rp) to the client and in both cases the server moves to a log-state (SL) and afterwards returns to the initial state (SQ) upon completion of logging (lg). So, the behaviour of Clients and Servers are synchronising via two actions: request and reply. The various timeouts are occurring independently.

The continuous time population model shown in Fig. 3 can be transformed into a discrete time population model by finding a suitable, high enough uniformisation rate to guarantee that both models have the same local states. Moreover, this rate is inversely proportional to the step size, which in turn should be small enough to guarantee convergence and absolute stability of the Euler method [26] that

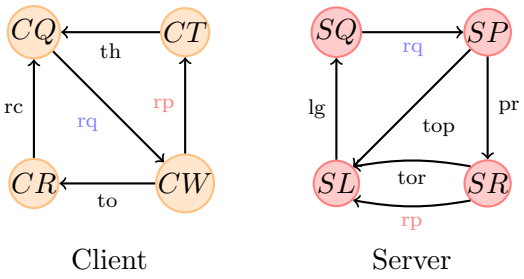


Figure 3: Client and Server process. *Client* rates: th=1, rp=100, rq=1, rc=100, to=0.01; *Server* rates: rq=1, pr=0.1, top=0.005, tor=0.005, rp=1, lg=10.

is used to approximate the solution of the differential equations of the fluid approximation of the population model.

First note that the rate of an action shared by two types of objects can never be higher than the rate of the objects that contribute to the synchronisation, and will be proportional to the normalised population size of the model. Therefore, we choose a uniformisation rate equal to the maximum total exit rate of any of the states of the objects. In the client-server case this maximum is 100.01 (the sum of the rates for Client actions *rp* and *to*). This may be an overestimation of the maximal rates, since the reply action of the Client is synchronised with that of the much slower Server (with reply rate 1). However, also the rate of the Client to recover (c_rc) is 100 and does not involve a synchronisation, so the uniformisation rate will need to be at least 100, which is the value we take.

Table 4 shows the translation of the continuous time model into a discrete time model in the probabilistic language of FlyFast. Note that, in the latter model, actions of different objects must be distinct. This is easily achieved by appropriate prefixing $c\_$ for Client actions and $s\_$ for Server ones, in the example). We derive the discrete time model of this example by taking the continuous time models of the individuals and replacing their rates with probabilities by dividing these original rates by the uniformisation rate $q = 100$. Note that in the simplified probability function of action c_rq we assume that the quantity CQ is always positive. This this is indeed the case is shown in Fig. 4(a).

We take this model as a first benchmark to compare the on-the-fly fluid model checking results with those of global fluid model checking presented in [6, 5]. A first check consists in comparing the transient occupancy measure of the two processes for the specification and parameter values shown in Table 4. The results are shown in Fig. 4. Given that we have taken the uniformisation rate equal to 100, we have that 1 time unit in continuous time corresponds to 100 time steps in the discrete model. In the initial state there are twice as many clients than servers. Each start in their initial state CQ (resp. SQ). The results for both analyses are very similar[7], and in turn show also good correspondence to results obtained with stochastic simulation of the model shown in [6, 5]. The close correspondence of the transient occupancy measures is also an indication that the step size used in the Euler forward method is small enough for leading to a satisfactory approximation for the time horizon of interest.

A further comparison concerns model checking results. In [5] fluid model checking is applied on a client process. One property concerns the probability mass of the CSL path formula $(CQ \vee CW)\ U^{\leq T}\ CR$, i.e. the probability that the client reaches the recover state CR after a time-out within T time-units, while passing only through client request CR and wait CW states being initially in the request state, for time bound T varying between 0 and 50 time-units. The result is shown in Fig. 5(a). Considering the scaling of time-units w.r.t. the number of steps and the translation of the CSL property in the corresponding PCTL property (see Def. 2), the results show a very good correspondence with those obtained via the fluid model checking procedure proposed in [5] on a

[7]There are some differences that are most likely to do with rescaling of the graphs in [5]. There the number of clients in state CQ appears to be initially 100%, whereas the proportions were supposed to be 2 clients for every server, so an initial percentage of 66%, as shown in Fig. 4(a) seems what was intended.

action c_rq : $min(\alpha_{c\_rq} * \mathsf{frc}\,(CQ), \alpha_{s\_rq} * \mathsf{frc}\,(SQ))/\mathsf{frc}\,(CQ)$;
action c_to : α_{c_to};
action c_rc : α_{c_rc};
action c_th : α_{c_th};
action c_rp : $min(\alpha_{c\_rp} * \mathsf{frc}\,(CW), \alpha_{s\_rp} * \mathsf{frc}\,(SR))/\mathsf{frc}\,(CW)$;

action s_rq : $min(\alpha_{c\_rq} * \mathsf{frc}\,(CQ), \alpha_{c\_rq} * \mathsf{frc}\,(SR))/\mathsf{frc}\,(SQ)$;
action s_pr : α_{s_pr};
action s_tor : α_{s_tor};
action s_top : α_{s_top};
action s_rp : $min(\alpha_{c\_rp} * \mathsf{frc}\,(CW), \alpha_{s\_rp} * \mathsf{frc}\,(SR))/\mathsf{frc}\,(SR)$;
action s_lg : α_{s_lg};

state $CQ\{c\_rq.CW\}$
state $CW\{c\_to.CR + c\_rp.CT\}$
state $CR\{c\_rc.CQ\}$
state $CT\{c\_th.CQ\}$

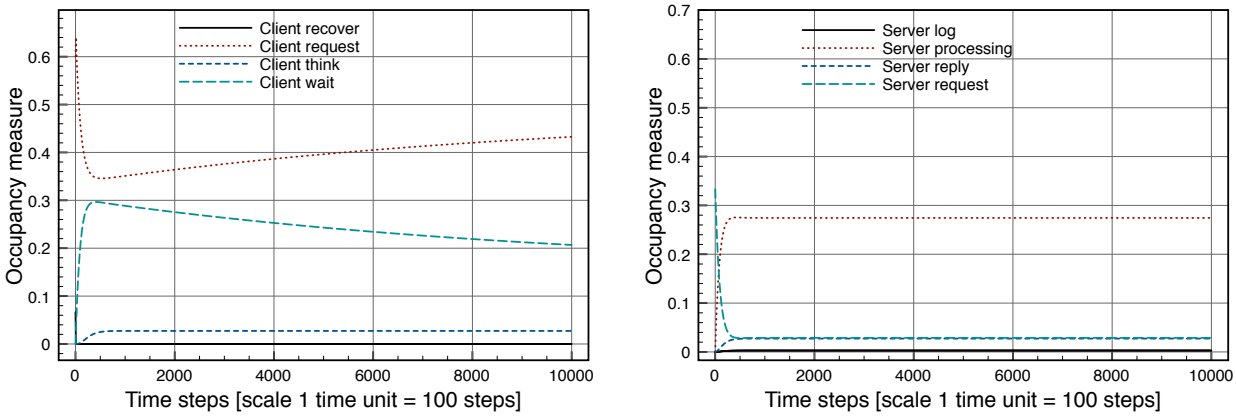
state $SQ\{s\_rq.SP\}$
state $SP\{s\_pr.SR + s\_top.SL\}$
state $SR\{s\_tor.SL + s\_rp.SL\}$
state $SL\{s\_lg.SQ\}$

system $mySystem =< CQ[1000], SQ[500] >$

Table 4: Client and Server specification in FlyFast with uniformisation rate $q = 100$ and the following values for the probabilities (not shown in the specification): $\alpha_{c\_th} = 1/q$, $\alpha_{c\_rp} = 100/q$, $\alpha_{c\_rq} = 1/q$, $\alpha_{c\_rc} = 100/q$, $\alpha_{c\_to} = 0.01/q$, $\alpha_{s\_rq} = 1/q$, $\alpha_{s\_pr} = 0.1/q$, $\alpha_{s\_top} = 0.005/q$, $\alpha_{s\_tor} = 0.005/q$, $\alpha_{s\_rp} = 1/q$, $\alpha_{s\_lg} = 10/q$.

continuous time model of the client. Fig. 5(b) shows model checking results of the same model, but for a fixed time bound $T = 5000$ and varying initial times $t0$ ranging from 0 to 2500 steps. The latter correspond to a time bound of 50 time-units in the original continuous model and initial times ranging from 0 to 25 time-units.

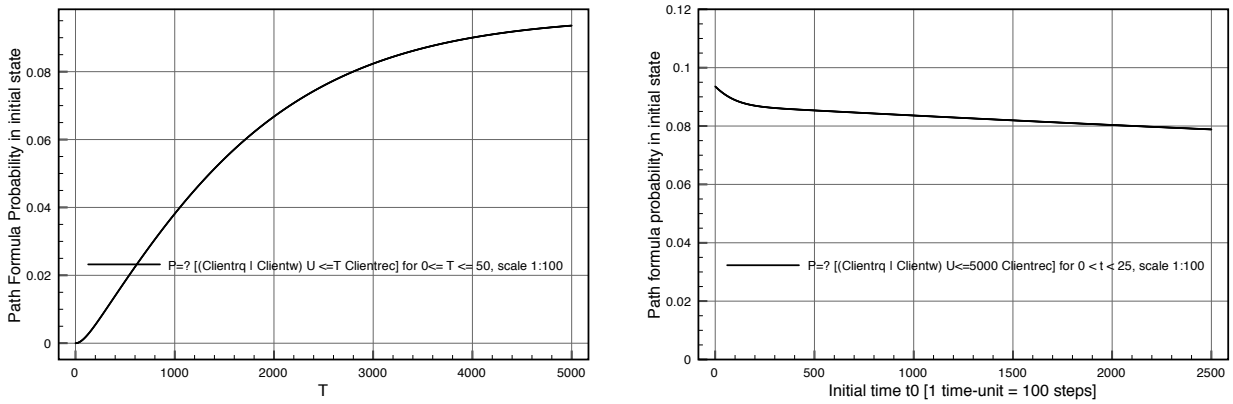
Fig. 5 shows results concerning the nested, time-dependent formula tt $U^T(P_{0.167}[\text{tt } U^{5000}\, CR])$, where CR denotes that a client timeout action c_to has occurred and the Client process has entered state CR. Nested formulas are computationally the most complex to analyse. The property is the PCTL version of the corresponding CSL nested property analysed by the global fluid model checking approach in [6, 5]. Since the discrete model has been obtained using a uniformisation rate of 100, the time bound of 50 time units in the CSL formula correspond now to 5000 steps in the discrete



(a) Transient occupancy measure of client states

(b) Transient occupancy measure of server states

Figure 4: Occupancy measure of client and server states



(a) Probability mass of $(CQ \vee CW)U^{\leq T}\,CR$ for varying time-bound T

(b) Probability mass of $(CQ \vee CW)U^{\leq 5000}\,CR$ for varying initial times $t0$

Figure 5: Model checking path formulae for Client process

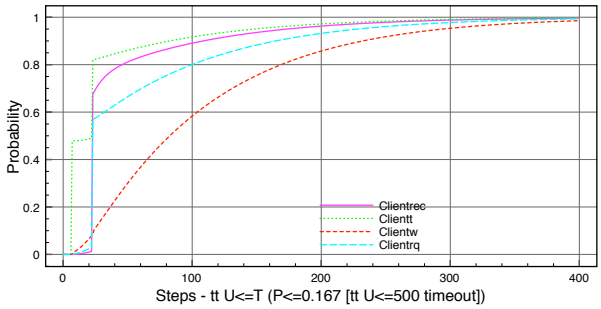
time setting. The results for T from 0 to 250 steps (corresponding to the first 2.5 time units in the continuous model) are shown in Fig. 6(c).

Note that the client-server model uses parameter values that differ several orders of magnitude, making the model very stiff and requiring a relatively high uniformisation rate. Fig. 6(a) shows FlyFast results for the same property, but for a model in which the recovery rate has been reduced to 1 (instead of 100), leading to a uniformisation rate of 10 (instead of 100). This makes the model checking much faster, but, of course, leads to different results. Note however, that three of the four curves are the same as in the previous case. This is because for those particular initial states of the individual Client process the rate of recovery is not involved. It is clear that for those curves the results correspond well also for higher values of T, including those used in the original model in [6, 5] (see Fig. 6(b) adopted from [5] to facilitate comparison).

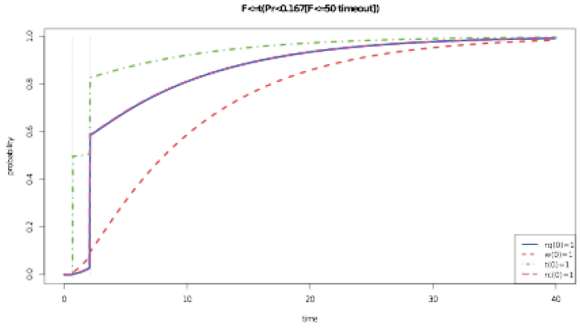
6 Related Work

Closest to our work is that by Bortolussi and Hillston [6, 5] presenting a technique for global Fluid Model Checking. We have briefly recalled some of the elements of this technique in Annex B and compared their model checking results with those obtained by our on-the-fly technique. To the best of our knowledge, the global Fluid Model Checking technique has not been fully implemented as yet, so we leave a more detailed comparison of the performance of both techniques to future work. Work on fluid model checking can also be found in [17] which uses in part similar techniques as [6]. On-the-fly probabilistic model checking for bounded PCTL has also been developed by Della Penna et al. [10] but they do not consider its use for on-the-fly fast mean field model checking as we did in [23].

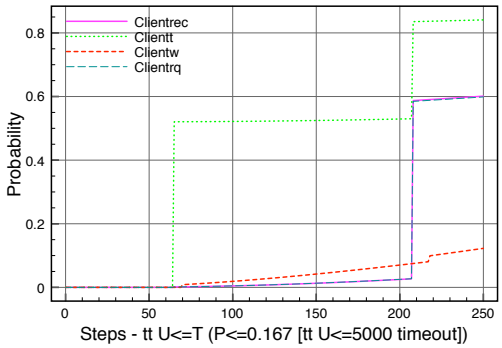
Another stochastic technique that addresses scalability is statistical model checking [15, 12, 28, 22]. In these techniques scalability is obtained by replacing a full state space search by a statistical analysis of a random set of executions of pre-established finite depth. This leads in many cases to a good scalability, however, the performance does remain dependent on the population size of the system under verification. The performance of techniques based on mean field and fluid approximation are independent of the population size, and thus scale to systems with an arbitrarily large population size. This is obtained, however, at the cost of abstracting from the individuality of the objects of which the system is composed. Rather than keeping track of the state of each individual, the fraction of objects in a particular local state at each time is taken as the system state, also called the 'occupancy measure'. The stochastic dynamics of the system approximates a deterministic function



(a) FlyFast results, model with reduced recovery rate and $q = 10$



(b) Original fluid model checking results of [6]



(c) FlyFast results for original model with $q = 100$ for the first 2.5 time units

Figure 6: FlyFast model checking results for a nested formula and for the Client initial states: CR, CT, CW and CQ

when the population size tends to infinite (this approach is correct under suitable conditions as shown by a number of convergence theorems both for the stochastic continuous time and the discrete time case [20, 21, 9, 25]). Such a deterministic function can be obtained as the solution of a set of ordinary differential equations (in the continuous time setting) or as difference equations (in the discrete time setting). A number of selected individuals can be analysed in combination with the approximation of the overall system behaviour. In this case the behaviour of the individual objects depends on the average "evolution" of the global system. This approach is justified by a corollary of the convergence theorems mentioned before and also known as *fast simulation* [9, 11].

7 Conclusions

We have illustrated an alternative way to perform fluid model checking of bounded CSL properties of individual entities in the context of CTMC population models. This alternative makes use of the prototype implementation of an on-the-fly fast mean field model checker FlyFast to check bounded PCTL formulas of individuals in the context of synchronous, discrete time DTMC population models. We have provided a correctness proof and obtained promising results compared to those available in the literature. Future work will consist in comparing the efficiency of our technique with that of global fluid model checking when prototype tools become fully available to us, to study further examples, to integrate the model and formula translations into FlyFast and to study the possibility of producing error bounds along with the analysis results.

8 Acknowledgements

This research has been partially funded by the EU projects QUANTICOL (nr. 600708) and ASCENS (nr. 257414), and the IT MIUR project CINA. Michele Loreti has two affiliations, his first affiliation is Dip. di Statistica, Informatica, Applicazioni, University of Florence, Florence, Italy, and his second affiliation is IMT Advanced Studies Lucca, Lucca, Italy. We would like to thank Luca Bortolussi for providing us with further details on the values of the parameters in the Client-Server model that made it possible to compare our results and for discussing fluid model checking.

References

[1] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Model checking Continuous Time Markov Chains. *ACM Transactions on Computational Logic*, 1(1):162–170, 2000.

[2] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-Checking Algorithms for Continuous-Time Markov Chains. *IEEE Transactions on Software Engineering. IEEE CS*, 29(6):524–541, 2003.

[3] M. Benaïm and J. Le Boudec. A class of mean field interaction models for computer and communication systems. *Performance Evaluation*, 65(11-12):823–838, 2008.

[4] G. Bhat, R. Cleaveland, and O. Grumberg. Efficient on-the-fly model checking for CTL*. In *LICS*, pages 388–397. IEEE Computer Society, 1995.

[5] L. Bortolussi and J. Hillston. Fluid model checking. *CoRR*, abs/1203.0920, 2012. Version 2 of Jan. 2013.

[6] L. Bortolussi and J. Hillston. Fluid model checking. In M. Koutny and I. Ulidowski, editors, *CONCUR*, volume 7454 of *LNCS*, pages 333–347. Springer-Verlag, 2012.

[7] L. Bortolussi, J. Hillston, D. Latella, and M. Massink. Continuous approximation of collective system behaviour: A tutorial. *Performance Evaluation*, 70(5):317 – 349, 2013.

[8] C. Courcoubetis, M. Vardi, P. Wolper, and M. Yannakakis. Memory-efficient algorithms for the verification of temporal properties. *Form. Methods Syst. Des.*, 1(2-3):275–288, 1992.

[9] R. Darling and J. Norris. Differential equation approximations for Markov chains. *Probability Surveys*, 5:37–79, 2008.

[10] G. Della Penna, B. Intrigila, I. Melatti, E. Tronci, and M. V. Zilli. Bounded probabilistic model checking with the mur*alpha* verifier. In A. J. Hu and A. K. Martin, editors, *FMCAD 2004*, volume 3312 of *LNCS*, pages 214–229. Springer, 2004.

[11] N. Gast and B. Gaujal. A mean field model of work stealing in large-scale systems. In V. Misra, P. Barford, and M. S. Squillante, editors, *SIGMETRICS*, pages 13–24. ACM, 2010.

[12] G. Guirado, T. Hérault, R. Lassaigne, and S. Peyronnet. Distribution, approximation and probabilistic model checking. In *PDMC 2005. LNCS, vol. 135*, pages 19–30. Springer, 2006.

[13] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.

[14] R. Hayden. *Scalable Performance Analysis of Massively Parallel Stochastic Systems*. PhD thesis, Imperial College London, April 2011.

[15] T. Hérault, R. Lassaigne, F. Magniette, and S. Peyronnet. Approximate probabilistic model checking. In *VMCAI04. LNCS, vol. 2937*, pages 73–84. Springer, 2004.

[16] J. Hillston. Fluid flow approximation of PEPA models. In *Proceedings of the Second International Conference on the Quantitative Evaluaiton of Systems (QEST 2005)*, pages 33–43, 2005.

[17] A. Kolesnichenko, P.-T. de Boer, A. Remke, and B. R. Haverkort. A logic for model-checking mean-field models. In *DSN*, pages 1–12. IEEE, 2013.

[18] S. Krantz and P. Harold. *A Primer of Real Analytic Functions (Second ed.)*. Birkhäuser, 2002.

[19] V. Kulkarni. *Modeling and Analysis of Stochastic Systems, First Edition*. Chapman & Hall, 1995.

[20] T. Kurtz. Solutions of ordinary differential equations as limits of pure jump Markov processes. *Journal of Applied Probability*, 7:49–58, 1970.

[21] T. Kurtz. Limit theorems for sequences of jump Markov processes approximating ordinary differential processes. *Journal of Applied Probability*, 8:244–356, 1971.

[22] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Proc. 23rd Int. Conf. on Computer Aided Verification (CAV'11)*, volume 6806 of *LNCS*, pages 585–591. Springer, Heidelberg, 2011.

[23] D. Latella, M. Loreti, and M. Massink. On-the-fly fast mean-field model-checking. In M. Abadi and A. Lluch-Lafuente, editors, *Trustworthy Global Computing - 8th International Symposium, TGC 2013, Buenos Aires, Argentina, August 30-31, 2013, Revised Selected Papers*, volume 8358 of *LNCS*, pages 297–314. Springer, 2013.

[24] D. Latella, M. Loreti, and M. Massink. On-the-fly probabilistic model-checking. In *Proceedings of the ICE 2014 Workshop*, 2014. To Appear in EPTCS.

[25] J.-Y. Le Boudec, D. McDonald, and J. Mundinger. A generic mean field convergence result for systems of interacting objects. In *QEST07*, pages 3–18. IEEE Computer Society Press, 2007. ISBN 978-0-7695-2883-0.

[26] R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations*. SIAM, 2007.

[27] M. Tribastone, S. Gilmore, and J. Hillston. Scalable differential analysis of process algebra models. *IEEE Trans. Software Eng.*, 38(1):205–219, 2012.

[28] H. L. S. Younes, M. Z. Kwiatkowska, G. Norman, and D. Parker. Numerical vs. statistical probabilistic model checking. *STTT*, 8(3):216–228, 2006.

A Lemmas and Auxiliary Definitions

First we present a few lemmas concerning Def. 1 and Def. 2, which will be useful in the proof of Theorem 1:

Lemma 1. *For matrices* $\mathbf{Q}$ *and* $\mathbf{U}$ *of Definition 1,* $s, s' \in \mathcal{S}$*, step size* $0 < d \in \mathbb{Q}$*, and* $k, n \in \mathbb{N}$ *the following holds:*

$$\left[\mathbf{I} \cdot \Pi_{i=k}^{k+n-1}[\mathbf{I} + d \cdot \mathbf{Q}(i \cdot d)]\right]_{s,s'} = [\mathbf{U}^n]_{(s,k),(s',k+n)}$$

Proof. By induction on n

Base case $(n = 0)$:

Trivial, since $\Pi_{i=k}^{k-1}[\mathbf{I} + d \cdot \mathbf{Q}(i \cdot d)] = \mathbf{I}$, by definition of Π and $\mathbf{U}^0 = \mathbf{I}$. Note that the resulting identity matrices have dimension $|\mathcal{S}| \cdot \lfloor \frac{b}{d} \rfloor \times |\mathcal{S}| \cdot \lfloor \frac{b}{d} \rfloor$, whereas the one within the product has dimension $|\mathcal{S}| \times |\mathcal{S}|$

Induction step $n + 1$:

$$\left[\mathbf{I} \cdot \Pi_{i=k}^{k+n-1+1}[\mathbf{I} + d \cdot \mathbf{Q}(i \cdot d)]\right]_{s,s'}$$

$=$ {Calculus}

$$\left[\mathbf{I} \cdot [\mathbf{I} + d \cdot \mathbf{Q}(k \cdot d)] \cdot \Pi_{i=k+1}^{k+n-1+1}[\mathbf{I} + d \cdot \mathbf{Q}(i \cdot d)]\right]_{s,s'}$$

$=$ {Calculus}

$$\left[[\mathbf{I} + d \cdot \mathbf{Q}(k \cdot d)] \cdot \mathbf{I} \cdot \Pi_{i=k+1}^{k+n-1+1}[\mathbf{I} + d \cdot \mathbf{Q}(i \cdot d)]\right]_{s,s'}$$

$=$ {Calculus}

$$\textstyle\sum_{s'' \in \mathcal{S}} [\mathbf{I} + d \cdot \mathbf{Q}(k \cdot d)]_{s,s''} \cdot \left[\mathbf{I} \cdot \Pi_{i=k+1}^{k+1+n-1}[\mathbf{I} + d \cdot \mathbf{Q}(i \cdot d)]\right]_{s'',s'}$$

$=$ {I.H.}

$$\textstyle\sum_{s'' \in \mathcal{S}} [\mathbf{I} + d \cdot \mathbf{Q}(k \cdot d)]_{s,s''} \cdot [\mathbf{U}^n]_{(s'',k+1),(s',k+1+n)}$$

$=$ {By definition of $\mathbf{U}$ when considering $n + k < \lfloor \frac{b}{d} \rfloor$}

$$\textstyle\sum_{s'' \in \mathcal{S}} [\mathbf{U}]_{(s,k),(s'',k+1)} \cdot [\mathbf{U}^n]_{(s'',k+1),(s',k+1+n)}$$

$=$ {Calculus}

$\left[\mathbf{U}^{n+1}\right]_{(s,k),(s',k+n+1)}$ □

We also introduce a notation for the transformation of states that satisfy a CSL formula Φ at time t into absorbing states at the level of an individual ICTMC Q-matrix.

Definition 4. *For CSL formula Φ and ICTMC model $\mathcal{M} = (\mathcal{S}, \alpha, \mathbf{Q})$ we let $\mathcal{M}[\Phi]$ be the ICTMC resulting from $\mathcal{M}$ by making all states satisfying Φ at time t absorbing at time t, i.e. $\mathcal{M}[\Phi] = (\mathcal{S}, \alpha, \mathbf{Q}')$ where, for $s \neq s'$*

$$\mathbf{Q}'_{s,s'}(t) = \begin{cases} 0, & \text{if } s, t \models_{\mathcal{M}} \Phi, \\ \mathbf{Q}_{s,s'}(t), & \text{otherwise.} \end{cases}$$

We use the same notation for the relevant infinitesimal generator matrices $\mathbf{Q}[\Phi]_{s,s'}(t) = \mathbf{Q}'_{s,s'}(t)$. Furthermore, for formula Φ, $k, n \in \mathbb{N}$, $i = k, \ldots, k+n$ and step size $0 < d \in \mathbb{Q}$ we let set $A_\Phi(i \cdot d) = \{s \in \mathcal{S} \mid s, i \cdot d \models_z \Phi\}$ and set $\tilde{A}_\Phi = \bigcup_{i=k}^{k+n} \{(s,i) \mid s \in A_\Phi(i \cdot d)\}$. With a some abuse of notation, we let $\mathbf{Q}[A_\Phi(i \cdot d)](i \cdot d) = \mathbf{Q}[\Phi](i \cdot d)$. Finally, we define $\mathbf{U}'$ as follows:

$$\mathbf{U}'_{(s,i),(s',i')} = \begin{cases} [\mathbf{I} + d \cdot \mathbf{Q}[A_\Phi(i \cdot d)](i \cdot d)]_{s,s'}, & \text{if } i' = i+1, s \notin A_\Phi(i \cdot d), \\ 1, & \text{if } i' = i, s' = s, s \in A_\Phi(i \cdot d), \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to see that $\mathbf{U}' = \mathbf{U}[\tilde{A}_\Phi]$, i.e. $\mathbf{U}$ where states in $\tilde{A}_\Phi$ are made absorbing (by making them self-loops with probability 1):

Lemma 2. *For all bounded CSL formulas Φ, $\mathbf{U}' = \mathbf{U}[\tilde{A}_\Phi]$*

Proof.

$$\begin{aligned}
& \mathbf{U}'_{(s,i),(s',i')} \\
= \quad & \{\text{By definition of } \mathbf{U}'\} \\
& \begin{cases} [\mathbf{I} + d \cdot \mathbf{Q}[A_\Phi(i \cdot d)](i \cdot d)]_{s,s'}, & \text{if } i' = i+1, s \notin A_\Phi(i \cdot d), \\ 1, & \text{if } i' = i, s' = s, s \in A_\Phi(i \cdot d), \\ 0, & \text{otherwise.} \end{cases} \\
= \quad & \{\text{By definition of } \mathbf{Q}[A_\Phi(i \cdot d)](i \cdot d)\} \\
& \begin{cases} [\mathbf{I} + d \cdot \mathbf{Q}]_{s,s'}, & \text{if } s \notin A_\Phi(i \cdot d) \text{ and } i' = i+1, \\ 1, & \text{if } s \in A_\Phi(i \cdot d) \text{ and } s' = s \text{ and } i' = i, \\ 0, & \text{otherwise.} \end{cases} \\
= \quad & \{s \in A_\Phi(i \cdot d) \text{ implies } (s,i) \in \tilde{A}_\Phi\} \\
& \mathbf{U}[\tilde{A}_\Phi]_{(s,i),(s',i')}
\end{aligned}$$

□

The following corollary of Lemma 2 is readily proven:

Corollary 1. *For matrices $\mathbf{Q}$ and $\mathbf{U}$ of Definition 1, $s, s' \in \mathcal{S}$, step size $d \in \mathbb{Q}$, and $k, n \in \mathbb{N}$ the following holds:*

$$\left[\mathbf{I} \cdot \Pi_{i=k}^{k+n} [\mathbf{I} + d \cdot \mathbf{Q}[A_\Phi(i \cdot d)](i \cdot d)]\right]_{s,s'} = \left[\mathbf{U}[\tilde{A}_\Phi]\right]^n_{(s,k),(s',k+n)}$$

Proof.

$$\Big[\mathbf{I}\cdot\Pi_{i=k}^{k+n}[\mathbf{I}+d\cdot\mathbf{Q}[A_\Phi(i\cdot d)](i\cdot d)]\Big]_{s,s'}$$

$=$ {Definition of $\mathbf{U}'$}

$$[\mathbf{U}']^n_{(s,k),(s',k+n)}$$

$=$ {Lemma 2: $\mathbf{U}'=\mathbf{U}[\tilde{A}_\Phi]$)}

$$\Big[\mathbf{U}[\tilde{A}_\Phi]\Big]^n_{(s,k),(s',k+n)}$$

□

B Proof of Main Theorem

In the proof of Theorem 1 we need some additional definitions and lemmas that concern the case of time-dependent reachability probability $\mathcal{P}_{\bowtie p}(\Phi_1\,\mathcal{U}^{\leq\tau}\,\Phi_2)$. The definition of this probability in bounded CSL is characterised as a piecewise transient analysis problem for global Fluid Model Checking as shown in [5] and involves two time-dependent sets, the set of goal states $G(t)$, i.e. the set of states that satisfy Φ_2 at time t, and the set of unsafe states $U(t)$, i.e. the set of states that satisfy neither Φ_1 nor Φ_2 at time t. These sets may change depending on the time at which formulas are evaluated and in the global Fluid Model Checking approach in [5] it is assumed that the time instances T_i in which such discontinuities in the sets occur are given *a priori*[8].

Let's assume k_I+1 of such points in time over a time interval of interest I of length τ starting at t, with $T_0=t$ and $T_{k_I+1}=t+\tau$, where t is the initial time and τ the time bound of the formula. Moreover, due to time-inhomogeneity, one also needs to deal in an appropriate way with the various time episodes in-between discontinuities and accumulate the probability of several sets of paths that satisfy the reachability property. This is taken care of by using two complementary mechanisms. One is to transform the $n\times n$ rate matrix $\mathbf{Q}(t)$ into a $2n\times 2n$ matrix $\bar{\mathbf{Q}}(t)$ making relevant states absorbing, but also by duplicating the local states $s\in\mathcal{S}$ into states $\bar{s}\in\bar{\mathcal{S}}$, so called 'shadow states', and redirecting transitions from states made absorbing to their shadow counterparts to accumulate the relevant reachability probability obtained during the various time episodes. The other is a transformation of the transient probability matrix $\bar{\mathbf{P}}(T_{i-1},T_i)$ at the end of each episode by multiplication by a suitable 0/1 matrix $\zeta(T_i)$. The operations and their justifications are treated in more detail in [5]. Here we only briefly recall their definitions. The matrix $\boldsymbol{\Upsilon}$ is defined as:

$$\boldsymbol{\Upsilon}(t,t+\tau)=\bar{\mathbf{P}}(t,T_1)\zeta(T_1)\bar{\mathbf{P}}(T_1,T_2)\zeta(T_2)...\zeta(T_{k_I})\bar{\mathbf{P}}(T_{k_I},t+\tau)$$

where $\bar{\mathbf{P}}(T_i,T_{i+1})$ is the transient probability matrix associated with the matrix $\bar{\mathbf{Q}}(t)$ of ICTMC $\bar{z}(t)$.

Definition 5. *The rate matrix $\bar{\mathbf{Q}}(t)$ is defined as follows:*

1. *for $\bar{s}_1\in\bar{\mathcal{S}}$ and any $s_2\in\bar{\mathcal{S}}\cup\mathcal{S}$, $\bar{q}_{\bar{s}_1,s_2}(t)=0$;*
2. *for s_1 a goal or unsafe state at t and all $s_2\in\bar{\mathcal{S}}\cup\mathcal{S}$, $\bar{q}_{s_1,s_2}(t)=0$;*
3. *for s_1 neither a goal nor unsafe state at t and $s_2\in\mathcal{S}\backslash G(t)$, $\bar{q}_{s_1,s_2}(t)=q_{s_1,s_2}(t)$, while $\bar{q}_{s_1,\bar{s}_2}(t)=0$;*
4. *for s_1 neither a goal nor unsafe state at t and $s_2\in G(t)$, $\bar{q}_{s_1,\bar{s}_2}(t)=q_{s_1,s_2}(t)$, while $\bar{q}_{s_1,s_2}(t)=0$*

Definition 6. *Matrix $\zeta(T_i)$ is the $2n\times 2n$ matrix:*

$$\zeta(T_i)=\begin{pmatrix}\zeta_W(T_i) & \zeta_G(T_i)\\ 0 & I\end{pmatrix}.$$

[8]The sets of goal states and unsafe states may change over time because the satisfaction of formulas depend on time (via the occupancy measure) in ICTMCs.

where I is the $n \times n$ identity matrix, $\zeta_W(T_i)$ the $n \times n$ matrix equal to 1 only on the diagonal elements corresponding to s_j that are neither a goal state nor an unsafe state both right before and right after time T_i, and 0 elsewhere. Furthermore, $\zeta_G(T_i)$ is the $n \times n$ matrix equal to 1 only on the diagonal elements corresponding to s_j that are neither a goal nor an unsafe state right before T_i, but become a goal state right after T_i.

The time-dependent reachability can then be defined as

$$\mathrm{Prob}^z(s, t, \Phi_1\, \mathcal{U}^{\leq \tau}\, \Phi_2) = \sum_{\bar{s}_1 \in \bar{\mathcal{S}}} \boldsymbol{\Upsilon}_{s,\bar{s}_1}(t, t+\tau) + \mathbf{1}\{s \in G(t)\}$$

Under the condition that the sets G and U converge robustly[9] to their limit sets, and that they do not have a discontinuity[10] at the same time for the same state, and that the rate functions involved in $\mathbf{Q}(t)$ are piecewise real analytic[11], it has been shown [5] that the time-varying reachability probability for process $Z^{(N)}$ converges to the limit process z.

In the discrete setting we need a discrete version of the matrix $\zeta(t)$, that we call η. Instead of checking whether a state changed its status (e.g. becoming a goal state) at exactly time instance T_i, in the discrete setting we need to check whether something changed from one step to the other around T_i.

Definition 7. *For $k \in \mathbb{N}$ the 0/1 matrix $\eta(k)$ is the $2n \times 2n$ matrix:*

$$\eta(k) = \begin{pmatrix} \eta_W(k) & \eta_G(k) \\ 0 & I \end{pmatrix}.$$

where I is the $n \times n$ identity matrix, $\eta_W(k)$ the $n \times n$ matrix equal to 1 only on the diagonal elements corresponding to s_j that are neither a goal state nor an unsafe state both at step $d \cdot k$ and $d \cdot (k+1)$, and 0 elsewhere. Furthermore, $\eta_G(k)$ is the $n \times n$ matrix equal to 1 only on the diagonal elements corresponding to s_j that are neither a goal nor an unsafe state at $d \cdot k$, but become a goal state at $d \cdot k + 1$.

Definition 8. *For all $0 < d \in \mathbb{Q}, b \in \mathbb{R}$ with $b > d$, $s, s' \in \mathcal{S} \cup \bar{\mathcal{S}}$, the matrix $\eta(k)$ defined as in Def. 7 and infinitesimal generator matrix $\bar{\mathbf{Q}}(t)$, the one step transition probability matrix $\bar{\mathbf{U}}$ is defined as follows:*

$$\bar{\mathbf{U}}_{(s,i),(s',i')} = \begin{cases} [[\mathbf{I} + d \cdot \bar{\mathbf{Q}}(i \cdot d)] \cdot \eta(i)]_{s,s'}, & \text{if } s \notin \bar{\mathcal{S}} \text{ and } i' = i+1, \\ 1, & \text{if } s \in \bar{\mathcal{S}}, s' = s \text{ and } i' = i, \\ 0, & \text{otherwise} \end{cases}$$

where the indexes of $\bar{\mathbf{U}}$ are assumed ordered as follows:

$$\begin{array}{l} (s_0,0),(s_1,0),\cdots,(s_n,0),(\bar{s}_0,0),(\bar{s}_1,0),\cdots,(\bar{s}_n,0), \\ (s_0,1),(s_1,1),\cdots,(s_n,1),(\bar{s}_0,1),(\bar{s}_1,1),\cdots,(\bar{s}_n,1), \\ \cdots, \\ (s_0,\lfloor \tfrac{b}{d} \rfloor),\cdots,(s_n,\lfloor \tfrac{b}{d} \rfloor),(\bar{s}_0,\lfloor \tfrac{b}{d} \rfloor),\cdots,(\bar{s}_n,\lfloor \tfrac{b}{d} \rfloor). \end{array}$$

Lemma 3. *For matrices $\bar{\mathbf{Q}}$ and $\bar{\mathbf{U}}$, of Definitions 5 and 8, $s, s' \in \mathcal{S} \cup \bar{\mathcal{S}}$, step size $0 < d \in \mathbb{Q}$, and $k, n \in \mathbb{N}$ the following holds:*

$$\left[\mathbf{I} \cdot \Pi_{i=k}^{k+n-1} [[\mathbf{I} + d \cdot \bar{\mathbf{Q}}(i \cdot d)] \cdot \eta(i)]\right]_{s,s'} = \left[\bar{\mathbf{U}}^n\right]_{(s,k),(s',k+n)}$$

[9] See [5] for a precise definition.

[10] By discontinuity here means that a state changes status at time t (e.g. from goal to unsafe, etc.).

[11] A function $f : I \to \mathbb{R}$, where I is an open subset of $\mathbb{R}$, is said to be analytic [18] in I if and only if for each point t_0 of I there is an open neighbourhood of I in which f coincides with its Taylor series expansion around t_0.

Proof. By induction on n.

Base case ($n = 0$):

Trivial, since $\Pi_{i=k}^{k-1}[[\mathbf{I} + d \cdot \bar{\mathbf{Q}}(i \cdot d)] \cdot \eta(i)] = \mathbf{I}$, by definition of Π and $\mathbf{U}^0 = \mathbf{I}$. Note that the resulting identity matrices have dimension $|\mathcal{S} \cup \bar{\mathcal{S}}| \cdot \lfloor \frac{b}{d} \rfloor \times |\mathcal{S} \cup \bar{\mathcal{S}}| \cdot \lfloor \frac{b}{d} \rfloor$, whereas the one within the product has dimension $|\mathcal{S} \cup \bar{\mathcal{S}}| \times |\mathcal{S} \cup \bar{\mathcal{S}}|$

Induction step $n + 1$:

$$\left[\mathbf{I} \cdot \Pi_{i=k}^{k+n-1+1}[[\mathbf{I} + d \cdot \bar{\mathbf{Q}}(i \cdot d)] \cdot \eta(i)]\right]_{s,s'}$$

$=$ {Calculus}

$$\left[\mathbf{I} \cdot [[\mathbf{I} + d \cdot \bar{\mathbf{Q}}(k \cdot d)] \cdot \eta(k)] \cdot \Pi_{i=k+1}^{k+n-1+1}[[\mathbf{I} + d \cdot \bar{\mathbf{Q}}(i \cdot d)] \cdot \eta(i)]\right]_{s,s'}$$

$=$ {Calculus}

$$\left[[[\mathbf{I} + d \cdot \bar{\mathbf{Q}}(k \cdot d)] \cdot \eta(k)] \cdot \mathbf{I} \cdot \Pi_{i=k+1}^{k+n-1+1}[[\mathbf{I} + d \cdot \bar{\mathbf{Q}}(i \cdot d)] \cdot \eta(i)]\right]_{s,s'}$$

$=$ {Calculus}

$$\textstyle\sum_{s'' \in \mathcal{S}}[[\mathbf{I} + d \cdot \bar{\mathbf{Q}}(k \cdot d)] \cdot \eta(k)]_{s,s''} \cdot \left[\mathbf{I} \cdot \Pi_{i=k+1}^{k+1+n-1}[[\mathbf{I} + d \cdot \bar{\mathbf{Q}}(i \cdot d)] \cdot \eta(i)]\right]_{s'',s'}$$

$=$ {I.H.}

$$\textstyle\sum_{s'' \in \mathcal{S}}[[\mathbf{I} + d \cdot \bar{\mathbf{Q}}(k \cdot d)] \cdot \eta(k)]_{s,s''} \cdot \left[\bar{\mathbf{U}}^n\right]_{(s'',k+1),(s',k+1+n)}$$

$=$ {By definition of $\bar{\mathbf{U}}$ when considering $n + k < \lfloor \frac{b}{d} \rfloor$}

$$\textstyle\sum_{s'' \in \mathcal{S}} \left[\bar{\mathbf{U}}\right]_{(s,k),(s'',k+1)} \cdot \left[\bar{\mathbf{U}}^n\right]_{(s'',k+1),(s',k+1+n)}$$

$=$ {Calculus}

$$\left[\bar{\mathbf{U}}^{n+1}\right]_{(s,k),(s',k+n+1)}$$

□

Lemma 4. *For matrices $\bar{\mathbf{U}}$ and $\mathbf{U}$, and set $R = \{(\bar{s}_1, n) | \bar{s}_1 \in \bar{\mathcal{S}} \wedge \lfloor \frac{t}{d} \rfloor \leq n \leq \lfloor (t + \tau)/d \rfloor\}$ we have that*

$$\textstyle\sum_{(\bar{s}_1,k) \in R} \left[\bar{\mathbf{U}}^{\lfloor \frac{\tau}{d} \rfloor}\right]_{(s,\lfloor \frac{t}{d} \rfloor),(\bar{s}_1,k)} + \mathbf{1}\{s \in G(\lfloor \frac{t}{d} \rfloor)\} \tag{1}$$

$=$

$$\textstyle\sum_{(s_1,k) \models_{\mathcal{T}_M(z,d,b)} \mathcal{T}_F[\![\Phi_2]\!]_d} \left[\mathbf{U}[\mathcal{T}_F[\![\neg\Phi_1]\!]_d \vee \mathcal{T}_F[\![\Phi_2]\!]_d]^{\lfloor \frac{\tau}{d} \rfloor}\right]_{(s,\lfloor \frac{t}{d} \rfloor),(s_1,k)} \tag{2}$$

Proof. First observe that if state s is a goal state at initial time t then the probability in Eq. (1) amounts to 1. In Eq. (2) this means that state $(s, \lfloor \frac{t}{d} \rfloor)$ has been turned into an absorbing state because it satisfies the PCTL equivalent of CSL formula Φ_2. This means that the only value for s_1 and k for which the elements indicated by coordinates $(s, \lfloor \frac{t}{d} \rfloor), (s_1, k)$ in $\mathbf{U}[\mathcal{T}_F[\![\neg\Phi_1]\!]_d \vee \mathcal{T}_F[\![\Phi_2]\!]_d]$ are not zero is $(s, \lfloor \frac{t}{d} \rfloor), (s, \lfloor \frac{t}{d} \rfloor)$ itself. The value of this element is equal to 1 (since it has been made absorbing).

If s is not a goal state at time t, then we need to collect the probability of all paths that reach a goal state at some time starting from s at time t. In the case of $\bar{\mathbf{U}}$ these probabilities are collected in the shadow states. These states are absorbing by definition of $\bar{\mathbf{Q}}$, that redirects transitions from s to its shadow for the period of time in which s is a goal, and matrix ζ that takes care that the

accumulated probabilities are not lost but appropriately accumulated. This is essentially the rôle of the identity matrix in the lower right quadrant of ζ. In $[\bar{\mathbf{U}}]^{\lfloor \frac{\tau}{d} \rfloor}$ the probabilities of the shadow states that can be reached from s in $\lfloor \tau/d \rfloor$ steps are present in the elements $(s, \lfloor \frac{t}{d} \rfloor), (\bar{s}_1, k)$. If $\bar{s}_1$ was not a goal state for some time steps k the corresponding probability for those time steps is zero. The same result is obtained in Eq. (2) by summing the probabilities corresponding to (s_1, k) that satisfy Φ_2 at time step k, meaning that it is a goal state at those steps. Turning (s_1, k) into an absorbing state when it satisfies Φ_2 means that the reachability probability of paths reaching s_1 from s at time t in $\lfloor \frac{\tau}{d} \rfloor$ steps is preserved in the same way as it is in $(\bar{s}_1, k)$. The reason why we do not need shadow states anymore is because the states $\mathcal{S}$ of the ICTMC have now been disambiguated by inserting the time step explicitly in the state.

For what concerns the unsafe states, one has to take care that the probability of paths passing through those states do not contribute to the total reachability probability. In the case of Eq. (1) this is taken care of by the matrix ζ, in particular by its left upper quadrant. The diagonal elements in η corresponding to states that are unsafe are set to zero. The effect of this is in Eq. (2) that in the product of $\bar{\mathbf{U}}$ the columns corresponding to unsafe states are always turned into zero, and so, the probability to leave unsafe states is effectively zero. The same effect can be reached by turning unsafe states into absorbing states. Also in that case, the probability to leave the state is zero. The final probability in the transient probability matrix corresponding to unsafe states is therefore not set to zero explicitly. But since only the probability of goal states count, the final reachability probability of interest is the same. □

Proof of Theorem 1. The proof of this theorem is by induction on the structure of the CSL formula Φ.

Proof.
Case a:
$s, t \models_z a$ if and only if $a \in \ell(s)$, by definition of $\models_z$. By definition of $\mathcal{T}_{Fd}$ we have $\mathcal{T}_F[\![a]\!]_d = a$, for all d. Furthermore, by definition of $\mathcal{T}_M$, $\mathcal{T}_M(z, d, b)$ has the same states as z, for all d and b. So $(s, \lfloor \frac{t}{d} \rfloor) \models_{\mathcal{T}_M(z,d,b)} a$.

Case $\neg\Phi$:
$s, t \models_z \neg\Phi$ if and only if $s, t \models_z \Phi$ does not hold, by definition of $\models_z$. Thus, by the induction hypothesis there exists N_0 such that for all $d \in D_\Phi, N \geq N_0$ and $b > \lceil \frac{|\Phi|}{d} \rceil$, $(s, \lfloor \frac{t}{d} \rfloor) \models_{\mathcal{T}_M(z,d,b)} \mathcal{T}_F[\![\Phi]\!]_d$ does not hold, i.e. using the fact that $|\neg\Phi| = |\Phi|$, we get that there exists N_0 such that for all $d \in D_\Phi, N \geq N_0$ and $b > \lceil \frac{|\neg\Phi|}{d} \rceil$, $(s, \lfloor \frac{t}{d} \rfloor) \models_{\mathcal{T}_M(z,d,b)} \mathcal{T}_F[\![\neg\Phi]\!]_d$, by definition of $\models_{\mathcal{T}_M(z,d,b)}$.

Case $\Phi_1 \vee \Phi_2$:
$s, t \models_z \Phi_1 \vee \Phi_2$ if and only if $s, t \models_z \Phi_1$ or $s, t \models_z \Phi_2$, by definition of $\models_z$. Thus, by the induction hypothesis, there exist N_1 such that for all $d \in D_\Phi$ and for all $N \geq N_1$ and $b > \lceil \frac{|\Phi_1|}{d} \rceil$, $(s, \lfloor \frac{t}{d} \rfloor) \models_{\mathcal{T}_M(z,d,b)}$ $\mathcal{T}_F[\![\Phi_1]\!]_d$ or there exist N_2 such that for all $d \in D_\Phi$ and for all $N \geq N_2$ and $b > \lceil \frac{|\Phi_2|}{d} \rceil$, $(s, \lfloor \frac{t}{d} \rfloor) \models_{\mathcal{T}_M(z,d,b)}$ $\mathcal{T}_F[\![\Phi_2]\!]_d$.

Thus, noting that $\lceil \frac{|\Phi_j|}{d} \rceil \leq \lceil \frac{|\Phi_1 \vee \Phi_2|}{d} \rceil$, for $j = 1, 2$, we get that there exist $N_0 \geq \max\{N_1, N_2\}$ such that for all $d \in D_\Phi$, and for all $N \geq N_0$ and $b > \lceil \frac{|\Phi_1 \vee \Phi_2|}{d} \rceil$ we have $(s, \lfloor \frac{t}{d} \rfloor) \models_{\mathcal{T}_M(z,d,b)} \mathcal{T}_F[\![\Phi_1 \vee \Phi_2]\!]_d$.

Case $\mathcal{P}_{\bowtie p}(\Phi_1 \, \mathcal{U}^{\leq \tau} \, \Phi_2)$:
$s, t \models_z \mathcal{P}_{\bowtie p}(\Phi_1 \, \mathcal{U}^{\leq \tau} \, \Phi_2)$ if and only if $\Pr\{\rho \in Paths_z(s, t) \mid \rho, t \models_z \Phi_1 \, \mathcal{U}^{\leq \tau} \, \Phi_2\} \bowtie p$, by definition of $\models_z$. Let us define $\text{Prob}^z(s, t, \Phi_1 \, \mathcal{U}^{\leq \tau} \, \Phi_2) = \Pr\{\rho \in Paths_z(s, t) \mid \rho, t \models_z \Phi_1 \, \mathcal{U}^{\leq \tau} \, \Phi_2\}$. The following shows that $\text{Prob}^z(s, t, \Phi_1 \, \mathcal{U}^{\leq \tau} \, \Phi_2)$ can be approximated by

$$\text{Prob}^{\mathcal{T}_M(z,d,b)}((s, \lfloor \frac{t}{d} \rfloor), \mathcal{T}_F[\![\Phi_1]\!]_d \, \mathcal{U}^{\leq \lfloor \frac{\tau}{d} \rfloor} \, \mathcal{T}_F[\![\Phi_2]\!]_d)$$

Let T_i with $i \in \{1, \cdots k_I\}$ be the k_I time instances in time interval I in which discontinuities occur in the sets of goal and unsafe states. We can then proceed with the following reasoning:

$\text{Prob}^z(s, t, \Phi_1\ \mathcal{U}^{\leq \tau}\ \Phi_2)$

$=$ {Expressing the reachability probability for CSL path formula $\Phi_1\ \mathcal{U}^{\leq \tau}\ \Phi_2$, as a piecewise transient analysis problem as for Fluid Model Checking [5], with T_i the given time instances at which discontinuities occur.}

$\sum_{\bar{s}_1 \in \bar{\mathcal{S}}} \boldsymbol{\Upsilon}_{s,\bar{s}_1}(t, t+\tau) + \mathbf{1}\{s \in G(t)\}$

$=$ {By Def. of $\boldsymbol{\Upsilon}$ and taking the T_i into consideration}

$\sum_{\bar{s}_1 \in \bar{\mathcal{S}}} [\bar{\mathbf{P}}(t, T_1)\zeta(T_1)\bar{\mathbf{P}}(T_1, T_2)\zeta(T_2) \cdots \zeta(T_{k_I})\bar{\mathbf{P}}(T_{k_I}, t+\tau)]_{s,\bar{s}_1} + \mathbf{1}\{s \in G(t)\}$

At this point, we approximate the transient probability matrices $\bar{\mathbf{P}}$ by those obtained from a forward Euler approximation with step size d. The step size is chosen sufficiently small such that the Euler approximation is stable and sufficiently accurate. We replace $\bar{\mathbf{P}}(t, T_1)$ by its Euler approximation[12] with step-size d, $\tilde{\mathbf{P}}(t, \lfloor T_1/d \rfloor d)$, and for $i \in \{1, \ldots, k_I\}$ the matrices $\bar{\mathbf{P}}(T_i, T_{i+1})$ by $\tilde{\mathbf{P}}(\lfloor T_i/d \rfloor d, \lfloor T_{i+1}/d \rfloor d)$ and finally $\bar{\mathbf{P}}(T_{k_I}, t+\tau)$ by $\tilde{\mathbf{P}}(\lfloor T_{k_I}/d \rfloor d, \lfloor (t+\tau)/d \rfloor d)$, where $\lfloor t/d \rfloor$ is the largest integer less than or equal to t/d. This approximation also implies that due to the fixed small step size d in the Euler approximation the end time of each transient probability may occur slightly before the time instances T_i in which a discontinuity in a state occurs, but by an amount strictly less than step size d. This is a possible source of small errors in the calculation of the total transient probability.

Also the function $\zeta(T_i)$ should take this switch to steps of size d into account. Instead of checking whether a state changed its status (e.g. becoming a goal state) at exactly time instance T_i, we should check whether something changed from one step to the other around T_i.

If we assume that the probability that $\lfloor T_i/d \rfloor d = T_i$ for each T_i is zero, then we have that $T_i - d < \lfloor T_i/d \rfloor d < T_i$. Let $j_i = \lfloor T_i/d \rfloor$ then the reachability probability can be approximated by:

$$\sum_{\bar{s}_1 \in \bar{\mathcal{S}}} [\tilde{\mathbf{P}}(t, j_1 d)\eta(j_1+1)(\prod_{m=1}^{m=k_I-1} \tilde{\mathbf{P}}(j_m d, j_{m+1} d)\eta(j_{m+1}+1))\tilde{\mathbf{P}}(j_{k_I} d, \lfloor (t+\tau)/d \rfloor d)]_{s,\bar{s}_1} + \mathbf{1}\{s \in G(t)\}$$

Using the definition of $\tilde{\mathbf{P}}$ and the fact that $\tilde{\mathbf{P}}(t, t) = I$, where I is the identity matrix, and moreover using the definition of $\bar{\mathbf{Q}}$ defined earlier, the above can be rewritten as:

$$\sum_{\bar{s}_1 \in \bar{\mathcal{S}}} \left[[I \cdot \prod_{i=\lfloor t/d \rfloor}^{j_1} [I + d\bar{\mathbf{Q}}(i.d)]].\eta(j_1+1).\left(\prod_{m=1}^{m=k_I-1} [I \cdot \prod_{i=j_m}^{j_{m+1}} [I + d\bar{\mathbf{Q}}(i.d)]]\eta(j_{m+1}+1) \right).[[I \cdot \prod_{i=j_{k_I}}^{\lfloor (t+\tau)/d \rfloor} [I + d\bar{\mathbf{Q}}(i.d)]] \right]_{s,\bar{s}_1} + \mathbf{1}\{s \in G(\lfloor \frac{t}{d} \rfloor)\}$$

The above formulation uses the fact that the discontinuity points T_i are known in advance. Computing such time instances a priori in the continuous time setting is however quite a delicate issue because this involves finding all zeros of an analytic function (i.e. the points in time in which a time bound of a formula is reached). This may in general not be decidable. See Bortolussi and Hillston [5] for a more detailed discussion of this issue. Note however, that in the new setting based on discrete time steps, we can in principle perform the η transformation after every step of the Euler approximation. In the case that none of the states change status at step i, the η transformation behaves as follows. First note that η in this case is almost the identity matrix. The submatrix $\eta_G(i)$ is the zero matrix, and the submatrix $\eta_W(i)$ has a 1 on the diagonal for all states that are neither unsafe nor goal states. For those that are unsafe or goal states the value is zero. But those states had already a special treatment. For goal states s the probability is collected in their shadow states $\bar{s}$ and no transitions lead the former, so the elements in the transient probability matrix corresponding to those states s

[12] See for example [19] p. 274.

can safely be set to zero. A similar argument holds for the unsafe states, because their probability should not be taken into account either. This means that we can rewrite the reachability probability as:

$$\sum_{\bar{s}_1 \in \bar{\mathcal{S}}} \left[[I \cdot \prod_{i=\lfloor t/d \rfloor}^{j_1} [I + d.\bar{\mathbf{Q}}(i.d)].\eta(i)]. \left(\prod_{m=1}^{m=k_I-1} [I \cdot \prod_{i=j_m}^{j_{m+1}} [I + d.\bar{\mathbf{Q}}(i.d)]\eta(i)] \right) .[[I \cdot \prod_{i=j_{k_I}}^{\lfloor (t+\tau)/d \rfloor} [I + d.\bar{\mathbf{Q}}(i.d)]] \right]_{s,\bar{s}_1} + \mathbf{1}\{s \in G(\lfloor \frac{t}{d} \rfloor)\}$$

Since we now check for discontinuities at every step, we can simplify the definition considering only the initial and end time of interest and not the discontinuities at T_i, giving:

$$\sum_{\bar{s}_1 \in \bar{\mathcal{S}}} \left[I \cdot \prod_{i=\lfloor t/d \rfloor}^{\lfloor (t+\tau)/d \rfloor} [I + d.\bar{\mathbf{Q}}(i.d)].\eta(i) \right]_{s,\bar{s}_1} + \mathbf{1}\{s \in G(\lfloor \frac{t}{d} \rfloor)\}$$

The above product matrix can equivalently be represented as a (possibly large) DTMC where the time steps are explicitly recorded in the states and which can be seen as a sort of 'unfolding' of the discretised ICTMC of the individual object. The dimension of this DTMC is $(|\mathcal{S} \cup \bar{\mathcal{S}}| \cdot \lfloor (t+\tau)/d \rfloor) \times (|\mathcal{S} \cup \bar{\mathcal{S}}| \cdot \lfloor (t+\tau)/d \rfloor)$. Let's denote this DTMC by $\bar{\mathbf{U}}$ since it includes also the shadow states $\bar{\mathcal{S}}$. Lemma 3 shows the correctness of this transformation. Furthermore, the states over which we need to sum are now $R = \{(\bar{s}_1, n) | \bar{s}_1 \in \bar{\mathcal{S}} \wedge \lfloor \frac{t}{d} \rfloor \leq n \leq \lfloor (t+\tau)/d \rfloor\}$:

$$\sum_{(\bar{s}_1,k) \in R} \left[\bar{\mathbf{U}}^{\lfloor \frac{\tau}{d} \rfloor} \right]_{(s,\lfloor \frac{t}{d} \rfloor),(\bar{s}_1,k)} + \mathbf{1}\{s \in G(\lfloor \frac{t}{d} \rfloor)\}$$

The next step is to show that this formulation is equivalent to the standard definition of reachability probability for ordinary DTMCs. In fact, at this point we can replace the method based on shadow states of the 'bookkeeping' procedures involved in the ICTMC setting by the usual method of computing transient probabilities of a DTMC in which the unsafe and goal states are made absorbing. The correspondence is formally shown in Lemma 4. Note that since the states also have the time step included, making a state absorbing in the DTMC setting therefore corresponds to an original ICTMC state being absorbing at a particular time.

$$\sum_{(s_1,k) \models_{\mathcal{T}_M(z,d,b)} \mathcal{T}_F[\![\Phi_2]\!]_d} \left[\mathbf{U}[\mathcal{T}_F[\![\neg\Phi_1]\!]_d \vee \mathcal{T}_F[\![\Phi_2]\!]_d]^{\lfloor \frac{\tau}{d} \rfloor} \right]_{(s,\lfloor \frac{t}{d} \rfloor),(s_1,k)}$$

This formulation directly reflects the standard way to compute the reachability probability in DTMC by reducing it to a transient analysis problem with one-step probability transition matrix $\mathbf{U}$ in which the unsafe and goal states are made absorbing. More formally:

$$\text{Prob}^{\mathcal{T}_M(z,d,b)}((s, \lfloor \frac{t}{d} \rfloor), \mathcal{T}_F[\![\Phi_1]\!]_d \, \mathcal{U}^{\leq \lfloor \frac{\tau}{d} \rfloor} \, \mathcal{T}_F[\![\Phi_2]\!]_d)$$

The latter bounded reachability probability for DTMC can also be computed in an *on-the-fly manner* as shown in [23]. The on-the-fly approach is convenient in this case, because the DTMC $\mathbf{U}$ may be very large due to the usually small step size d that needs to be considered in order for the difference equations involved in the Euler approximation to be stable and converging. With the on-the-fly approach only that part of the state space is generated from the high-level model that is strictly necessary to check the property of interest. □