Fluid Approximation for Stochastic Model Checking

Jane Hillston
joint work with Luca Bortolussi

Laboratory for Foundations of Computer Science
University of Edinburgh
(Dipartimento di Matematica ed Informatica
University of Trieste)

24th October 2012

The main story

Fluid methods have been very successful as an approximate description of the collective (average) behaviour of Stochastic (Process Algebra) models, even for moderately sized systems.

They have also been applied to estimate the passage times.

R.A.Hayden, A.Stefanek, J.T.Bradley. *Fluid computation of passage-time distributions in large Markov models.* TCS 2012.

Can we use them to query stochastic models and estimate more complex stochastic properties?

Stated otherwise:

Can we do fluid model checking?

The main story

Fluid methods have been very successful as an approximate description of the collective (average) behaviour of Stochastic (Process Algebra) models, even for moderately sized systems.

They have also been applied to estimate the passage times.

R.A.Hayden, A.Stefanek, J.T.Bradley. *Fluid computation of passage-time distributions in large Markov models.* TCS 2012.

Can we use them to query stochastic models and estimate more complex stochastic properties?

Stated otherwise:

Can we do fluid model checking?

The main story

Fluid methods have been very successful as an approximate description of the collective (average) behaviour of Stochastic (Process Algebra) models, even for moderately sized systems.

They have also been applied to estimate the passage times.

R.A.Hayden, A.Stefanek, J.T.Bradley. *Fluid computation of passage-time distributions in large Markov models.* TCS 2012.

Can we use them to query stochastic models and estimate more complex stochastic properties?

Stated otherwise:

Can we do fluid model checking?

The goal of this work

We will consider collective systems, composed of many interacting agents of one or more classes.

We will focus on questions related to the behaviour of an individual agent in the system (or of a small fixed collection of agents).

Examples

There are many examples in which this can be interesting:

- Estimate performance metrics in network models, from the point of view of the single user/single server.
- Ecological models, when one is interested at the survival chances of an individual.
- ...

The goal of this work

We will consider collective systems, composed of many interacting agents of one or more classes.

We will focus on questions related to the behaviour of an individual agent in the system (or of a small fixed collection of agents).

Examples

There are many examples in which this can be interesting:

- Estimate performance metrics in network models, from the point of view of the single user/single server.
- Ecological models, when one is interested at the survival chances of an individual.
- ...

The goal of this work

We will consider collective systems, composed of many interacting agents of one or more classes.

We will focus on questions related to the behaviour of an individual agent in the system (or of a small fixed collection of agents).

Examples

There are many examples in which this can be interesting:

- Estimate performance metrics in network models, from the point of view of the single user/single server.
- Ecological models, when one is interested at the survival chances of an individual.
- ...

Outline

Population models — introduction to notation

Individuals

We have N individuals $Y_i^{(N)} \in S$, $S = \{1, 2, \ldots, n\}$ in the system (can have multiple classes).

System variables

$X_j^{(N)} = \sum_{i=1}^{N} \mathbf{1}\{Y_i^{(N)} = j\}$, and $\mathbf{X}^{(N)} = (X_1^{(N)}, \ldots, X_n^{(N)})$

Dynamics (system level)

$\mathbf{X}^{(N)}$ is a CTMC with transitions $\tau \in \mathcal{T}$:

$$\tau\text{: } \mathbf{X}^{(N)} \text{ to } \mathbf{X}^{(N)} + \mathbf{v}_\tau \text{ at rate } r_\tau^{(N)}(\mathbf{X})$$

Population models — introduction to notation

Individuals

We have N individuals $Y_i^{(N)} \in S$, $S = \{1, 2, \ldots, n\}$ in the system (can have multiple classes).

System variables

$X_j^{(N)} = \sum_{i=1}^{N} \mathbf{1}\{Y_i^{(N)} = j\}$, and $\mathbf{X}^{(N)} = (X_1^{(N)}, \ldots, X_n^{(N)})$

Dynamics (system level)

$\mathbf{X}^{(N)}$ is a CTMC with transitions $\tau \in \mathcal{T}$:

$$\tau: \mathbf{X}^{(N)} \text{ to } \mathbf{X}^{(N)} + \mathbf{v}_\tau \text{ at rate } r_\tau^{(N)}(\mathbf{X})$$

Population models — introduction to notation

Individuals

We have N individuals $Y_i^{(N)} \in S$, $S = \{1, 2, \ldots, n\}$ in the system (can have multiple classes).

System variables

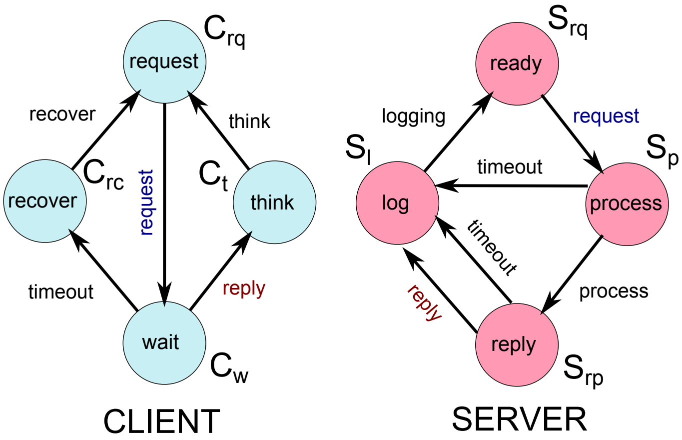
$X_j^{(N)} = \sum_{i=1}^{N} \mathbf{1}\{Y_i^{(N)} = j\}$, and $\mathbf{X}^{(N)} = (X_1^{(N)}, \ldots, X_n^{(N)})$

Dynamics (system level)

$\mathbf{X}^{(N)}$ is a CTMC with transitions $\tau \in \mathcal{T}$:

$$\tau\text{: } \mathbf{X}^{(N)} \text{ to } \mathbf{X}^{(N)} + \mathbf{v}_\tau \text{ at rate } r_\tau^{(N)}(\mathbf{X})$$

Example: client server interaction



Example: client server interaction

Variables

- 4 variables for the client states: C_{rq}, C_w, C_{rc} C_t.
- 4 variables for the server states: S_{rq}, S_p, S_{rp} S_l.

Transitions

There are 7 transition in totals. We use synchronisation as in PEPA:

- request: $(\cdot, \mathbf{1}_{C_w,S_p} - \mathbf{1}_{C_{rq},S_{rq}}, kr \cdot min(C_{rq}, S_{rq}))$
- reply: $(\cdot, \mathbf{1}_{C_t,S_l} - \mathbf{1}_{C_w,S_{rp}}, min(k_w C_w, k_{rp} S_{rp}))$
- timeout: $(\cdot, \mathbf{1}_{C_{rc}} - \mathbf{1}_{C_w}, k_{to} C_w)$
- ...

Scaling Conditions

Scaling assumptions

- We have a sequence $\mathbf{X}^{(N)}$ of population CTMC, for increasing total population N.
- We normalize such models, dividing variables by N: $\overline{\mathbf{X}}^{(N)} = \frac{\mathbf{X}}{N}$
- for each $\tau \in \mathcal{T}^{(N)}$, the normalized update is $\bar{\mathbf{v}} = \mathbf{v}/N$ and the rate function is $\bar{r}_\tau(\overline{\mathbf{X}}^{(N)}) = N f_\tau(\overline{\mathbf{X}}^{(N)})$ (density dependence).

Fluid ODE

The fluid ODE is $\dot{\mathbf{x}} = F(\mathbf{x})$, where

$$F(\mathbf{x}) = \sum_{\tau \in \mathcal{T}} \mathbf{v}_\tau f_\tau(\mathbf{x})$$

Fluid approximation theorem

Hypothesis

- $\overline{\mathbf{X}}^{(N)}(t)$: a sequence of normalized population CTMC, residing in $E \subset \mathbb{R}^n$
- $\exists \mathbf{x_0} \in S$ such that $\overline{\mathbf{X}}^{(N)}(0) \rightarrow \mathbf{x_0}$ in probability (initial conditions)
- $\mathbf{x}(t)$: solution of $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = F(\mathbf{x})$, $\mathbf{x}(0) = \mathbf{x_0}$, residing in E.

Theorem

For any finite time horizon $T < \infty$, it holds that:

$$\mathbb{P}(\sup_{0 \leq t \leq T} ||\overline{\mathbf{X}}^{(N)}(t) - \mathbf{x}(t)|| > \varepsilon) \rightarrow 0.$$

T.G.Kurtz. *Solutions of ordinary differential equations as limits of pure jump Markov processes.* Journal of Applied Probability, 1970.

Single Agent Asymptotic Behaviour

Dynamics of individuals

Focus on a single individual $Y_h^{(N)}$, which is a stochastic process on $S = \{1, \ldots, n\}$ (but NOT Markov!).

Let $Q^{(N)}(\mathbf{x})$ be the "infinitesimal generator matrix" of $Y_h^{(N)}$:
$\mathbb{P}\{Y_h^{(N)}(t + dt) = j \mid Y_h^{(N)}(t) = i, \overline{\mathbf{X}}^{(N)}(t) = \mathbf{x}\} = q_{i,j}^{(N)}(\mathbf{x})dt.$

Suppose $Q^{(N)}(\mathbf{x}) \to Q(\mathbf{x})$

R. Darling, J. Norris. Differential equation approximations for Markov chains. *Probability Surveys*, 2008.

We suppose that as the population increases the transition rates of the individual tend to the transition rates of an individual dependent instead on the mean field.

Single Agent Asymptotic Behaviour

Dynamics of individuals

Focus on a single individual $Y_h^{(N)}$, which is a stochastic process on $S = \{1, \ldots, n\}$ (but NOT Markov!).

Let $Q^{(N)}(\mathbf{x})$ be the "infinitesimal generator matrix" of $Y_h^{(N)}$:
$\mathbb{P}\{Y_h^{(N)}(t + dt) = j \mid Y_h^{(N)}(t) = i, \overline{\mathbf{X}}^{(N)}(t) = \mathbf{x}\} = q_{i,j}^{(N)}(\mathbf{x})dt$.

Suppose $Q^{(N)}(\mathbf{x}) \rightarrow Q(\mathbf{x})$

R. Darling, J. Norris. Differential equation approximations for Markov chains. *Probability Surveys*, 2008.

We suppose that as the population increases the transition rates of the individual tend to the transition rates of an individual dependent instead on the mean field.

Fast Simulation

Asymptotic behaviour of $Z_k^{(N)}$

Let $\mathbf{x}(t)$ be the solution of the fluid ODE, and assume to be under the hypothesis of Kurtz theorem.

Let $z_k(t)$ be the time inhomogeneous-CTMC on S^k defined by the following infinitesimal generator (for any $h = 1, \ldots, k$):

$$\mathbb{P}\{z_k(t+dt) = (\ldots, j, \ldots) \mid z_k(t+dt) = (\ldots, i, \ldots)\} = q_{i,j}(\mathbf{x}(t))dt$$

Theorem (Fast simulation theorem)

For any $T < \infty$, $\mathbb{P}\{Z_k^{(N)}(t) \neq z_k(t), t \leq T\} \to 0.$

R. Darling, J. Norris. Differential equation approximations for Markov chains. *Probability Surveys*, 2008.

Fast Simulation

Asymptotic behaviour of $Z_k^{(N)}$

Let $\mathbf{x}(t)$ be the solution of the fluid ODE, and assume to be under the hypothesis of Kurtz theorem.

Let $z_k(t)$ be the time inhomogeneous-CTMC on S^k defined by the following infinitesimal generator (for any $h = 1, \ldots, k$):

$$\mathbb{P}\{z_k(t+dt) = (\ldots, j, \ldots) \mid z_k(t+dt) = (\ldots, i, \ldots)\} = q_{i,j}(\mathbf{x}(t))dt$$

Theorem (Fast simulation theorem)

For any $T < \infty$*,* $\mathbb{P}\{Z_k^{(N)}(t) \neq z_k(t), t \leq T\} \rightarrow 0.$

R. Darling, J. Norris. Differential equation approximations for Markov chains. *Probability Surveys*, 2008.

Client-Server example

Single client

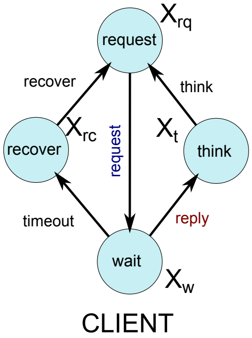
$Y^{(N)} \in \{rq, w, t, rc\}$

Rates of $Z_1^{(N)}$

- request: $\frac{1}{C_{rq}^{(N)}} k_r \min(C_{rq}^{(N)}, S_{rq}^{(N)})$
- reply: $\frac{1}{C_w^{(N)}} \min(k_w C_w^{(N)}, k_{rp} S_{rp}^{(N)})$
- timeout: k_{to}; recover: k_{rc}

Rates of z_1

- request: $k_r \min(1, \frac{s_{rq}(t)}{c_{rq}(t)})$
- reply: $\min(k_w, k_{rp} \frac{s_{rp}(t)}{c_w(t)})$
- timeout: k_{to}; recover: k_{rc}



The idea

Approximate the behaviour of an agent Z in the system using the time-inhomogeneous Markov chain z.

Model check temporal logic formulae on z.
In this work, we consider CSL logic with time bounded temporal operators.

Outline of results

- A model checking algorithm for CSL on time-inhomogeneous CTMC (ICTMC).
- Investigation of its decidability.
- Convergence results (asymptotic correctness for large N).

L. Bortolussi, J. Hillston. Fluid Model Checking. CONCUR 2012 and Corr arXiv, 1203.0920.

The idea

Approximate the behaviour of an agent Z in the system using the time-inhomogeneous Markov chain z.

Model check temporal logic formulae on z.
In this work, we consider CSL logic with time bounded temporal operators.

Outline of results

- A model checking algorithm for CSL on time-inhomogeneous CTMC (ICTMC).
- Investigation of its decidability.
- Convergence results (asymptotic correctness for large N).

L. Bortolussi, J. Hillston. Fluid Model Checking. CONCUR 2012 and Corr arXiv, 1203.0920.

Logic: (time-bounded) CSL

Paths of a stochastic process on $\mathcal{S}$

A path of $V(t)$ is a sequence $\sigma = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots$, with non null probability of jumping from s_i to s_{i+1}, etc. Denote with $\sigma@t$ the state at time t.

States of $V(t)$ are labelled by atomic propositions $a_1, a_2, \ldots$

(Time-Bounded) Continuous Stochastic Logic

$$\phi = a \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \mathcal{P}_{\bowtie p}(\phi_1\, U^{[T_1,T_2]}\, \phi_2)$$

Logic: (time-bounded) CSL

Paths of a stochastic process on $\mathcal{S}$

A path of $V(t)$ is a sequence $\sigma = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots$, with non null probability of jumping from s_i to s_{i+1}, etc. Denote with $\sigma@t$ the state at time t.

States of $V(t)$ are labelled by atomic propositions $a_1, a_2, \ldots$

(Time-Bounded) Continuous Stochastic Logic

$$\phi = a \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \mathcal{P}_{\bowtie p}(\phi_1\ U^{[T_1,T_2]}\ \phi_2)$$

Logic: (time-bounded) CSL

Paths of a stochastic process on $\mathcal{S}$

A path of $V(t)$ is a sequence $\sigma = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots$, with non null probability of jumping from s_i to s_{i+1}, etc. Denote with $\sigma@t$ the state at time t.

States of $V(t)$ are labelled by atomic propositions $a_1, a_2, \ldots$

(Time-Bounded) Continuous Stochastic Logic

$$\phi = a \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \mathcal{P}_{\bowtie p}(\phi_1\, U^{[T_1,T_2]}\, \phi_2)$$

Satisfiability for CSL

$$s, t_0 \models \mathcal{P}_{\bowtie p}(\phi_1 \, U^{[T_1,T_2]} \, \phi_2) \quad \text{iff} \quad \mathbb{P}\{\sigma \mid \sigma, t_0 \models \phi_1 \, U^{[T_1,T_2]} \, \phi_2\} \bowtie p.$$

$$\sigma, t_0 \models \phi_1 \, U^{[T_1,T_2]} \phi_2 \quad \text{iff} \quad \exists \bar{t} \in [t_0 + T_1, t_0 + T_2] \text{ such that } \sigma @ \bar{t} \models \phi_2 \text{ and } \forall t_0 \leq t < \bar{t}, \sigma @ t \models \phi_1.$$

Satisfiability for CSL

$$s, t_0 \models \mathcal{P}_{\bowtie p}(\phi_1\, U^{[T_1,T_2]}\, \phi_2) \quad \text{iff} \quad \mathbb{P}\{\sigma \mid \sigma, t_0 \models \phi_1\, U^{[T_1,T_2]}\, \phi_2\} \bowtie p.$$

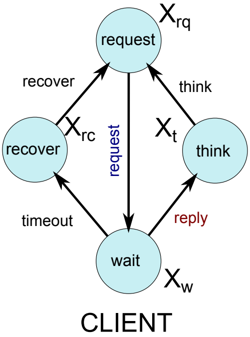
$$\sigma, t_0 \models \phi_1\, U^{[T_1,T_2]} \phi_2 \quad \text{iff} \quad \exists \bar{t} \in [t_0 + T_1, t_0 + T_2] \text{ such that } \sigma @ \bar{t} \models \phi_2 \text{ and } \forall t_0 \leq t < \bar{t},\ \sigma @ t \models \phi_1.$$

Satisfiability for CSL

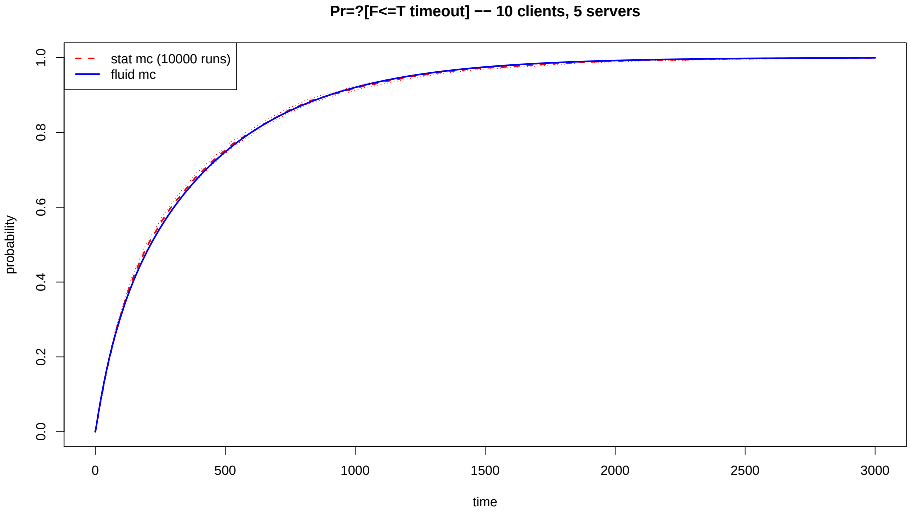
$$s, t_0 \models \mathcal{P}_{\bowtie p}(\phi_1 \, U^{[T_1,T_2]} \, \phi_2) \quad \text{iff} \quad \mathbb{P}\{\sigma \mid \sigma, t_0 \models \phi_1 \, U^{[T_1,T_2]} \, \phi_2\} \bowtie p.$$

$$\sigma, t_0 \models \phi_1 \, U^{[T_1,T_2]} \phi_2 \quad \text{iff} \quad \exists \bar{t} \in [t_0 + T_1, t_0 + T_2] \text{ such that } \sigma @ \bar{t} \models \phi_2 \text{ and } \forall t_0 \leq t < \bar{t}, \sigma @ t \models \phi_1.$$

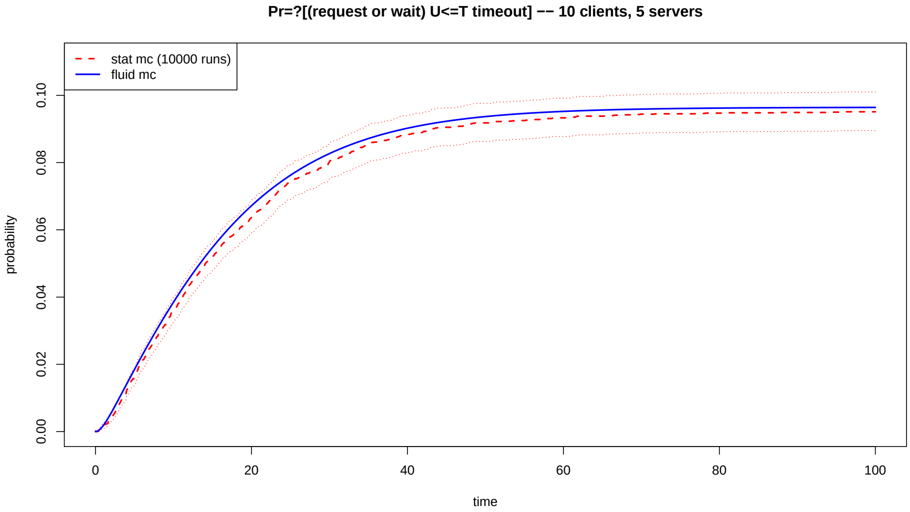
Client-Server example



Client-Server: $\mathcal{P}_{=?}(F^{\leq T} a_{timeout})$



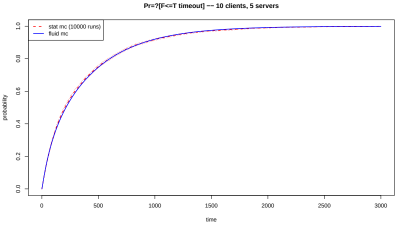
Client-Server: $\mathcal{P}_{=?}(a_{request} \vee a_{wait} U^{\leq T} a_{timeout})$

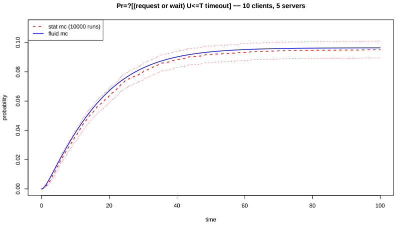


Client-Server: computational cost

Computational cost

The cost of the fluid system is independent of N.
For this example (10 clients - 5 servers) it is $\sim$100 times faster than the simulation-based approach (which increases linearly with N).





CSL model checking for CTMC

Consider a CTMC with state space S and time varying rates $Q = Q(t)$.
Focus on the formula

$$\mathcal{P}_{\bowtie p}\,(\phi_1\; U^{[0,T]}\; \phi_2)$$

Time-homogeneous CTMC

For time-homogeneous case, we can check this formula by computing, for each state $s \in S$, the probability of paths satisfying $\phi_1\; U^{[0,T]}\; \phi_2$ deciding if this probability is $\bowtie p$.

This is done via transient analysis on the chain in which $\neg\phi_1$ and ϕ_2 states are made absorbing.

Time-homogeneity $\Rightarrow$ we can run each transient analysis from time $t_0 = 0$ even if we have nested until formulae.

CSL model checking for CTMC

Consider a CTMC with state space S and time varying rates $Q = Q(t)$.

Focus on the formula

$$\mathcal{P}_{\bowtie p}\left(\phi_1 \, U^{[0,T]} \, \phi_2\right)$$

Time-homogeneous CTMC

For time-homogeneous case, we can check this formula by computing, for each state $s \in S$, the probability of paths satisfying $\phi_1 \, U^{[0,T]} \, \phi_2$ deciding if this probability is $\bowtie p$.

This is done via transient analysis on the chain in which $\neg\phi_1$ and ϕ_2 states are made absorbing.

Time-homogeneity $\Rightarrow$ we can run each transient analysis from time $t_0 = 0$ even if we have nested until formulae.

CSL model checking for CTMC

Consider a CTMC with state space S and time varying rates $Q = Q(t)$.

Focus on the formula

$$\mathcal{P}_{\bowtie p}\,(\phi_1\ U^{[0,T]}\ \phi_2)$$

This is no longer true in time-inhomogeneous CTMCs, as the probability of an until formula depends on the time at which we evaluate it.

The truth value of ϕ in a state s
depends on the time t at which we evaluate it!

This causes problems when we consider nested until formulae.

Model Checking ICTMC — related work

The CTMC $z_n^{(N)}$ is time-inhomogeneous.

Model checking for ICTMC

- Model checking Hennessy Milner Logics (ICTMC with piecewise constant rates)
- Model checking LTL (time unbounded operators, requires asymptotic regularity of rates).
- Model checking against DTA specification (not for ICTMC, can possibly be extended)

- J.P. Katoen, A. Mereacre. Model Checking HML on Piecewise-Constant Inhomogeneous Markov Chains. FORMATS 2008.
- T. Chen, T. Han, J.P. Katoen, A. Mereacre: LTL Model Checking of Time-Inhomogeneous Markov Chains. ATVA 2009.
- T. Chen, T. Han, J.P. Katoen, A. Mereacre: Model Checking of Continuous-Time Markov Chains Against Timed Automata Specifications. Logical Methods in Computer Science 7, 2011.

CSL model checking for ICTMC

Consider a ICTMC with state space S and rates $Q = Q(t)$.

$$\mathcal{P}_{\bowtie p}(\phi_1 \; U^{[0,T]} \; \phi_2)$$

CSL model checking for ICTMC

Consider a ICTMC with state space S and rates $Q = Q(t)$.

$$\mathcal{P}_{\bowtie p}(\phi_1 \, U^{[0,T]} \, \phi_2)$$

This can be model checked using transient analysis to solve the following reachability problem:

What is the probability of reaching a ϕ_2-state within time T without entering a $\neg\phi_1$-state?

Kolmogorov forward and backward equations

Let $\Pi(t_1, t_2) = (\pi_{s_i,s_j}(t_1, t_2))_{i,j}$ be the probability matrix giving the probability of being in state s_j at time t_2, given that we are in state s_i at time t_1.

The Kolmogorov forward and backward equations describe the time evolution of $\Pi(t_1, t_2)$ as a function of t_1 and t_2 respectively.

Kolmogorov forward and backward equations

$$\frac{\partial \Pi(t_1, t_2)}{\partial t_2} = \Pi(t_1, t_2)Q(t_2) \qquad \frac{\partial \Pi(t_1, t_2)}{\partial t_1} = -Q(t_1)\Pi(t_1, t_2).$$

Kolmogorov forward and backward equations

Let $\Pi(t_1, t_2) = (\pi_{s_i,s_j}(t_1, t_2))_{i,j}$ be the probability matrix giving the probability of being in state s_j at time t_2, given that we are in state s_i at time t_1.

The Kolmogorov forward and backward equations describe the time evolution of $\Pi(t_1, t_2)$ as a function of t_1 and t_2 respectively.

Kolmogorov forward and backward equations

$$\frac{\partial \Pi(t_1, t_2)}{\partial t_2} = \Pi(t_1, t_2)Q(t_2) \qquad \frac{\partial \Pi(t_1, t_2)}{\partial t_1} = -Q(t_1)\Pi(t_1, t_2).$$

Kolmogorov forward and backward equations

Let $\Pi(t_1, t_2) = (\pi_{s_i,s_j}(t_1, t_2))_{i,j}$ be the probability matrix giving the probability of being in state s_j at time t_2, given that we are in state s_i at time t_1.

The Kolmogorov forward and backward equations describe the time evolution of $\Pi(t_1, t_2)$ as a function of t_1 and t_2 respectively.

Kolmogorov forward and backward equations

$$\frac{\partial \Pi(t_1, t_2)}{\partial t_2} = \Pi(t_1, t_2) Q(t_2) \qquad \frac{\partial \Pi(t_1, t_2)}{\partial t_1} = -Q(t_1)\Pi(t_1, t_2).$$

Time-Dependent reachability probability

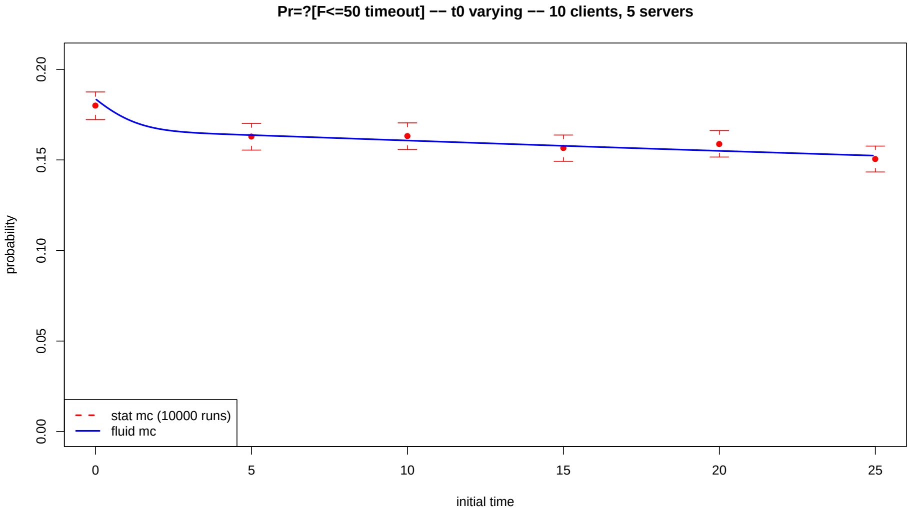
If we want to compute $\Pi(t, t+T)$ as a function of t, let $t' = t + T$ (initial conditions, $\Pi(0, T)$):

$$\frac{\mathrm{d}\Pi(t, t+T)}{\mathrm{d}t} = \frac{\partial \Pi(t, t')}{\partial t'} \frac{\partial t'}{\partial t} + \frac{\partial \Pi(t, t')}{\partial t}$$

$$\frac{\mathrm{d}\Pi(t, t+T)}{\mathrm{d}t} = \Pi(t, t+T) Q(t+T) - Q(t) \Pi(t, t+T)$$

Then, $P_{\phi_1 U^{[0,T]} \phi_2}(s, t) =$ is equal to $\sum_{s' \models \phi_2} \Pi_{\neg\phi_1 \wedge \phi_2}(t, t+T)_{s,s'}$.

Client Server: $\mathcal{P}_{=?}F^{\leq T}a_{timeout}$ as a function of t_0



Time-dependent truth

- When computing the truth value of an until formula, we obtain a time dependent value $\mathbf{T}(\phi, s, t)$ in each state.
- When we consider nested temporal operators, we need to take this into account.
- The problem is that in this case the topology of goal and unsafe states in the CTMC can change in time.

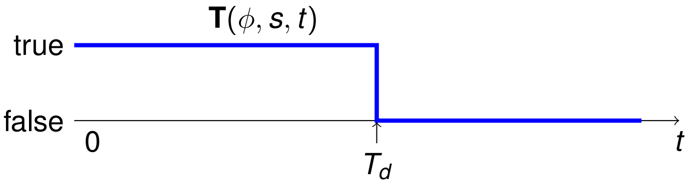
Time-dependent truth

- When computing the truth value of an until formula, we obtain a time dependent value **T**(ϕ, s, t) in each state.
- When we consider nested temporal operators, we need to take this into account.
- The problem is that in this case the topology of goal and unsafe states in the CTMC can change in time.

Time-dependent truth

- When computing the truth value of an until formula, we obtain a time dependent value $\mathbf{T}(\phi, s, t)$ in each state.
- When we consider nested temporal operators, we need to take this into account.
- The problem is that in this case the topology of goal and unsafe states in the CTMC can change in time.

Time dependent truth: $\Box^{\leq T}\phi$

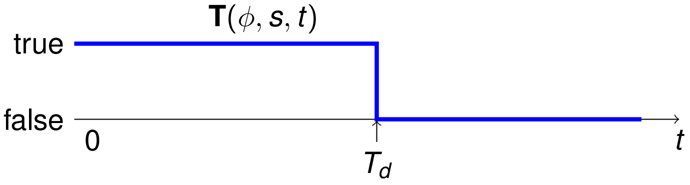


At discontinuity times, changes in topology introduce discontinuities in the probability values.

But...

Discontinuities happen at specific and fixed time instants. We can solve Kolmogorov equations piecewise!

Time dependent truth: $\Box^{\leq T}\phi$

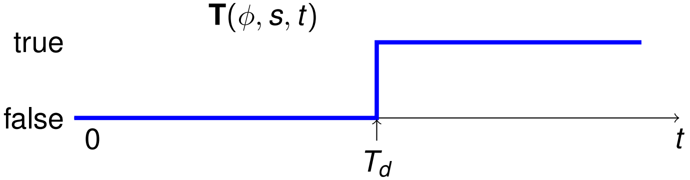


More precisely

Write $\Pi(t, t+T) = \zeta_{T_d}(\Pi(t, T_d))\Pi(T_d, t+T)$, and derive Kolmogorov equations by applying chain rule.

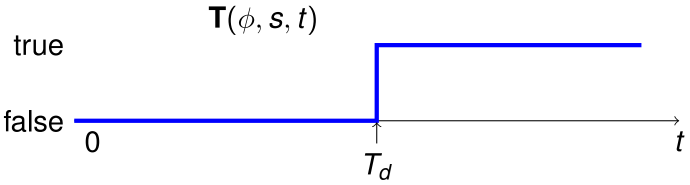
Here ζ_{T_d} sets to zero all entries $\pi_{s,s'}$, such that s is a $\neg\phi$-state before time T_d and s' is a $\neg\phi$-state either before of after time T_d.

Time dependent truth: $F^{\leq T}\phi$



- State s becomes a goal state at time T_d.
- If we are in state s at time T_d^- (without having reached a ϕ state before), then we are suddenly in a ϕ-state at time T_d^+.
- At time T_d we need to add $\pi_{s',s}(t, T_d)$ to the reachability probability from each state s'.
- This introduces discontinuities in the reachability probability.
- At each discontinuity event, we also have to appropriately re-route the Q matrix.

Time dependent truth: $F^{\leq T}\phi$

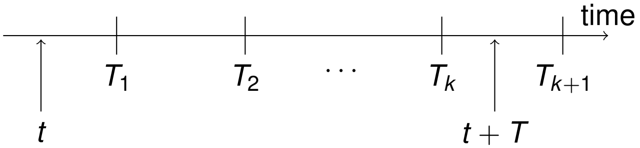


Pragmatically

In both cases, at each discontinuity event, we have to appropriately re-route the Q matrix.

For bookkeeping reasons, we need to add $|S|$ additional sink variables that collect the probability of reaching a ϕ-state within T time units from time t to $t + T$.

k discontinuities $T_1, \ldots, T_k$ in $[t, t+T]$



The generic Chapman-Kolmogorov equation

$\Pi(t, t+T) = \Pi_1(t, T_1)\zeta(T_1)\Pi_2(T_1, T_2)\zeta(T_2)\cdots\zeta(T_k)\Pi_{k+1}(T_k, t+T)$.
$\zeta(T_j)$ apply the appropriate bookkeeping operations to deal with changes in the topology of absorbing states.

- We can compute $\Pi(t, t+T)$ by an ODE obtained by derivation and application of chain rule.
- In advancing time, when we hit a discontinuity point (from below or above), the structure of the previous equation changes: integration has to be stopped and restarted.

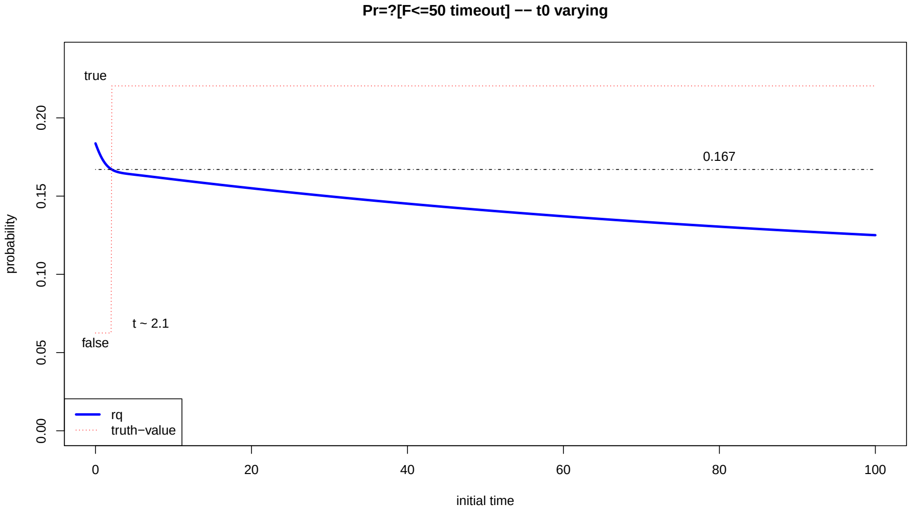
The Algorithm (sketched)

Proceed bottom-up on the parse tree of a formula.
Case $\mathbf{T}(\mathcal{P}_{\bowtie p}(\phi_1 U^{[0,T]} \phi_2), t)$:

- Compute $\mathbf{T}(\phi_1, t)$ and $\mathbf{T}(\phi_2, t)$
- Let $T_1, \ldots, T_m$ be all the discontinuity points of $\mathbf{T}(\phi_1, t)$ and $\mathbf{T}(\phi_2, t)$ up to a final time T_f.
- Compute $\Pi(T_i, T_i + 1)$ for each i
- Compute $\Pi(0, T)$ using generalized CK equations
- Integrate $\frac{d}{dt}\Pi(t, t + T)$ up to T_f.
- Return $\mathbf{T}(\mathcal{P}_{\bowtie p}(\phi_1 U^{[0,T]} \phi_2), t) = \Pi(t, t + T) \bowtie p$.

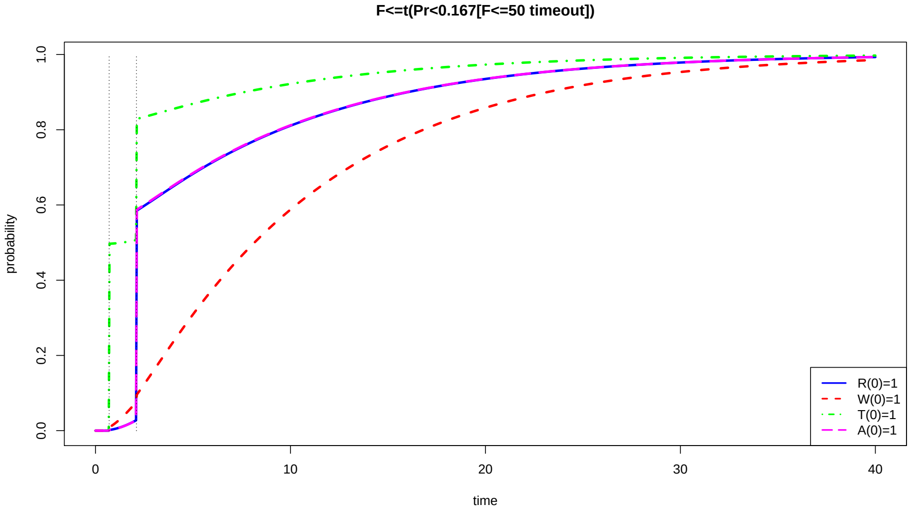
The use of Kolmogorov equations is feasible if the state space is small.
This is usually the case for single agent mean field models.

Client-Server: $F^{\leq T}(P_{<0.167}(F^{\leq 50}\, timeout))$



$P_{<0.167}(F^{\leq 50}\, timeout)$ from state *rq* of client.

Client-Server: $F^{\leq T}(P_{<0.167}(F^{\leq 50}\, timeout))$



Time-dependent until probability

There are two issues which we need to consider:

- Numerical stability of the integration of the forward+backward equation.
- Number of zeros of the function $P_{\phi_1 U^{[T,T']}\phi_2}(s,t) - p$: is it always finite, if we restrict our attention to a compact time interval $[0, T_{max}]$? Can it be infinite?

Time-dependent until probability

There are two issues which we need to consider:

- Numerical stability of the integration of the forward+backward equation.
- Number of zeros of the function $P_{\phi_1 U^{[T,T']} \phi_2}(s,t) - p$: is it always finite, if we restrict our attention to a compact time interval $[0, T_{max}]$? Can it be infinite?

Decidability

Number of zeros of $P(t) - p$

- We want that this equation has a finite number of solutions in each $[0, T]$.
- We can enforce this by requiring rate functions of ICTMC to be piecewise real-analytic functions.

Decidability

Decidability

- We need algorithms to solve ODEs with error guarantee (interval analysis).
- We need to find zeros of function $P(s,t) - p$ (root finding).
- To answer the CSL query for main until formulae, we need to know if $P(s,0) \bowtie p$ (zero test).
- It is not known if root finding and zero test are decidable.

Decidability

Decidability

- We need algorithms to solve ODEs with error guarantee (interval analysis).
- We need to find zeros of function $P(s, t) - p$ (root finding).
- To answer the CSL query for main until formulae, we need to know if $P(s, 0) \bowtie p$ (zero test).
- It is not known if root finding and zero test are decidable.

Decidability

Decidability

- We need algorithms to solve ODEs with error guarantee (interval analysis).
- We need to find zeros of function $P(s, t) - p$ (root finding).
- To answer the CSL query for main until formulae, we need to know if $P(s, 0) \bowtie p$ (zero test).
- It is not known if root finding and zero test are decidable.

Decidability

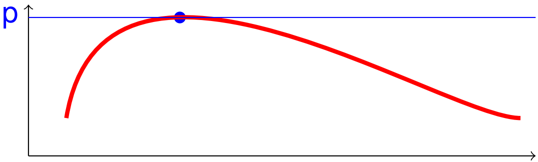
Decidability

- We need algorithms to solve ODEs with error guarantee (interval analysis).
- We need to find zeros of function $P(s,t) - p$ (root finding).
- To answer the CSL query for main until formulae, we need to know if $P(s,0) \bowtie p$ (zero test).
- It is not known if root finding and zero test are decidable.

Decidability

Decidability

- We need algorithms to solve ODEs with error guarantee (interval analysis).
- We need to find zeros of function $P(s, t) - p$ (root finding).
- To answer the CSL query for main until formulae, we need to know if $P(s, 0) \bowtie p$ (zero test).
- It is not known if root finding and zero test are decidable.



Decidability

Theorem (Quasi-decidability)

Let $\phi = \phi(\mathbf{p})$ *be a CSL formula, with constants* $\mathbf{p} = (p_1, \ldots, p_k) \in [0,1]^k$ *appearing in until formulae.*

The CSL model checking for ICTMC problem is decidable for $\mathbf{p} \in E$*, where* E *is an open subset of* $[0,1]^k$*, of measure 1.*

Convergence of CSL truth

- We considered also convergence of CSL properties: properties that are true in z_k are eventually true in $Z_k^{(N)}$?
- Convergence suffers from similar issues to decidability: tangential zeros and $P(s, 0) = p$ can create problems.

Theorem (Asymptotic correctness)

Let $\phi = \phi(\mathbf{p})$ be a CSL formula, with constants $\mathbf{p} = (p_1, \ldots, p_k) \in [0, 1]^k$ appearing in until formulae.

Then, for $\mathbf{p} \in E$, an open subset of $[0, 1]^k$ of measure 1, there exists N_0 such that $\forall N \geq N_0$

$$s, 0 \models_{Z_k^{(N)}} \phi \Leftrightarrow s, 0 \models_{z_k} \phi.$$

Conclusions

- We presented an application of mean field theory to model check properties of single agents in a large population.
- We focussed on CSL, providing a method to model check CSL formulae versus time-inhomogeneous CTMC.
- We provided convergence results that guarantee almost surely consistence of the method.

Conclusions

- We presented an application of mean field theory to model check properties of single agents in a large population.
- We focussed on CSL, providing a method to model check CSL formulae versus time-inhomogeneous CTMC.
- We provided convergence results that guarantee almost surely consistence of the method.

Conclusions

- We presented an application of mean field theory to model check properties of single agents in a large population.
- We focussed on CSL, providing a method to model check CSL formulae versus time-inhomogeneous CTMC.
- We provided convergence results that guarantee almost surely consistence of the method.

Future work

- Investigate the use of error bounds for mean field convergence to provide a (rough) estimate of the error.
- Investigate better the “individual to population” relationship (average behaviour, estimates for probability)
- Include rewards, next operator, and steady state, when possible.
- Working implementation
- Consider other logics on single agents (e.g. MTL)