The Benefits of Sometimes Not Being Discrete

Jane Hillston
LFCS, University of Edinburgh

2nd September 2014

Outline

Outline

The Discrete World View

As computer scientists we generally take a discrete view of the world.

The Discrete World View

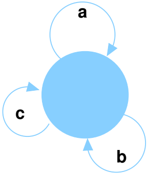
As computer scientists we generally take a discrete view of the world.

This is particularly true when we want to reason about the behaviour of systems, as most formalisms are built upon notions of states and transitions.

The Discrete World View

As computer scientists we generally take a discrete view of the world.

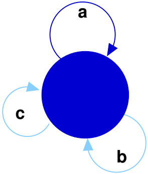
This is particularly true when we want to reason about the behaviour of systems, as most formalisms are built upon notions of states and transitions.



The Discrete World View

As computer scientists we generally take a discrete view of the world.

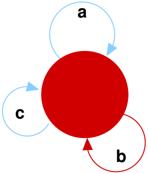
This is particularly true when we want to reason about the behaviour of systems, as most formalisms are built upon notions of states and transitions.



The Discrete World View

As computer scientists we generally take a discrete view of the world.

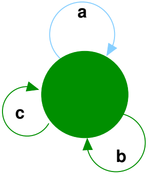
This is particularly true when we want to reason about the behaviour of systems, as most formalisms are built upon notions of states and transitions.



The Discrete World View

As computer scientists we generally take a discrete view of the world.

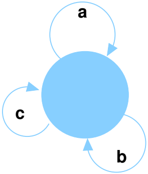
This is particularly true when we want to reason about the behaviour of systems, as most formalisms are built upon notions of states and transitions.



The Discrete World View

As computer scientists we generally take a discrete view of the world.

This is particularly true when we want to reason about the behaviour of systems, as most formalisms are built upon notions of states and transitions.



Various formalisms have been designed for capturing such behaviour.

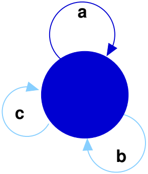
Process Algebra

- Models consist of agents which engage in actions.

$$\alpha.P$$

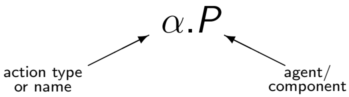
action type or name → α

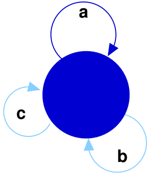
agent/ component → P



Process Algebra

- Models consist of agents which engage in actions.

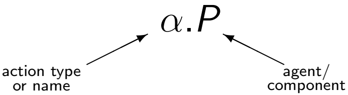


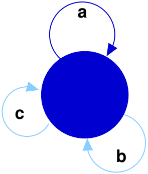


- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

Process Algebra

- Models consist of agents which engage in actions.



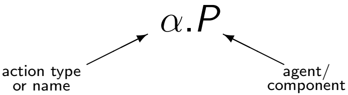


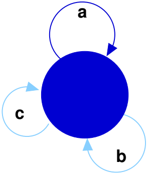
- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

Process algebra
model

Process Algebra

- Models consist of agents which engage in actions.



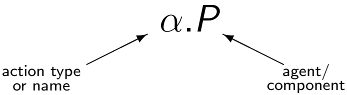


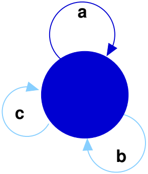
- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

Process algebra model —SOS rules→

Process Algebra

- Models consist of agents which engage in actions.





- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

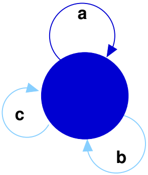
Process algebra model —SOS rules→ Labelled transition system

Process Algebra

- Models consist of agents which engage in actions.

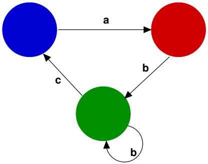
$$\alpha.P$$

action type or name → α ; agent/ component → P



- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

Process algebra model —SOS rules→ Labelled transition system



Quantitative Modelling

Performance modelling aims to construct models of the dynamic behaviour of systems in order to support the efficient and equitable sharing of resources. Availability and reliability modelling consider the dynamic behaviour of systems with failures and breakdowns.

Quantitative Modelling

Performance modelling aims to construct models of the dynamic behaviour of systems in order to support the efficient and equitable sharing of resources. Availability and reliability modelling consider the dynamic behaviour of systems with failures and breakdowns.

Markovian-based discrete event models have been applied to computer systems since the mid-1960s and communication systems since the early 20th century.

Quantitative Modelling

Performance modelling aims to construct models of the dynamic behaviour of systems in order to support the efficient and equitable sharing of resources. Availability and reliability modelling consider the dynamic behaviour of systems with failures and breakdowns.

Markovian-based discrete event models have been applied to computer systems since the mid-1960s and communication systems since the early 20th century.

Originally queueing networks were primarily used to construct models, and sophisticated analysis techniques were developed.

Quantitative Modelling

Performance modelling aims to construct models of the dynamic behaviour of systems in order to support the efficient and equitable sharing of resources. Availability and reliability modelling consider the dynamic behaviour of systems with failures and breakdowns.

Markovian-based discrete event models have been applied to computer systems since the mid-1960s and communication systems since the early 20th century.

Originally queueing networks were primarily used to construct models, and sophisticated analysis techniques were developed.

These techniques are no longer widely applicable for expressing the dynamic behaviour observed in distributed systems with concurrent behaviour.

Formal Approaches to Quantitative Modelling

The size and complexity of real systems makes the direct construction of discrete state models costly and error-prone.

Formal Approaches to Quantitative Modelling

The size and complexity of real systems makes the direct construction of discrete state models costly and error-prone.

For the last three decades there has been substantial interest in applying formal modelling techniques enhanced with information about timing and probability.

Formal Approaches to Quantitative Modelling

The size and complexity of real systems makes the direct construction of discrete state models costly and error-prone.

For the last three decades there has been substantial interest in applying formal modelling techniques enhanced with information about timing and probability.

From these high-level system descriptions the underlying mathematical model (Continuous Time Markov Chain (CTMC)) can be automatically generated.

Formal Approaches to Quantitative Modelling

The size and complexity of real systems makes the direct construction of discrete state models costly and error-prone.

For the last three decades there has been substantial interest in applying formal modelling techniques enhanced with information about timing and probability.

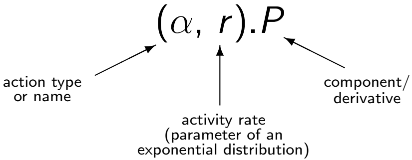
From these high-level system descriptions the underlying mathematical model (Continuous Time Markov Chain (CTMC)) can be automatically generated.

Primary examples include:

- Stochastic Petri Nets and
- Stochastic/Markovian Process Algebras.

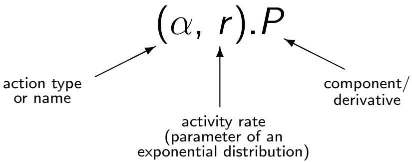
Stochastic Process Algebra

- Models are constructed from components which engage in activities.



Stochastic Process Algebra

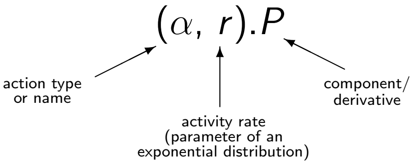
- Models are constructed from components which engage in activities.



- The language is used to generate a CTMC for performance modelling.

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

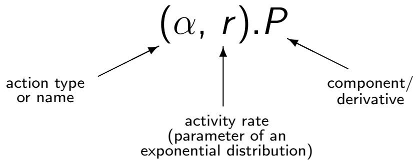


- The language is used to generate a CTMC for performance modelling.

SPA
MODEL

Stochastic Process Algebra

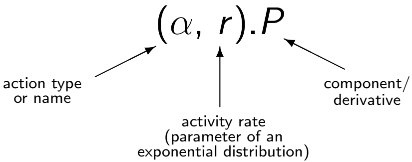
- Models are constructed from components which engage in activities.



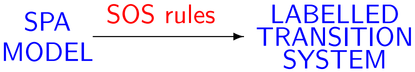
- The language is used to generate a CTMC for performance modelling.

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

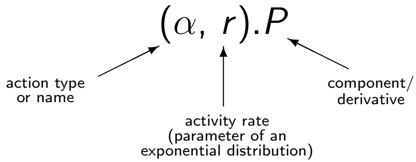


- The language is used to generate a CTMC for performance modelling.



Stochastic Process Algebra

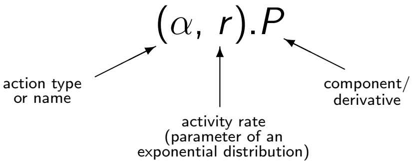
- Models are constructed from components which engage in activities.



- The language is used to generate a CTMC for performance modelling.

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



- The language is used to generate a CTMC for performance modelling.

Integrated analysis

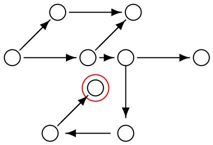
Qualitative verification can now be complemented by quantitative verification.

Integrated analysis

Qualitative verification can now be complemented by quantitative verification.

Reachability analysis

How long will it take for the system to arrive in a particular state?

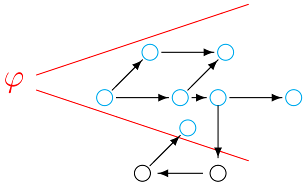


Integrated analysis

Qualitative verification can now be complemented by quantitative verification.

Model checking

Does a given property φ hold within the system with a given probability?

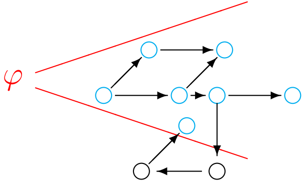


Integrated analysis

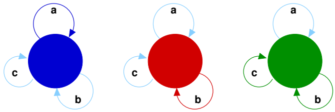
Qualitative verification can now be complemented by quantitative verification.

Model checking

For a given starting state
how long is it until
a given property φ holds?



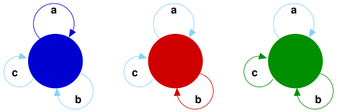
Solving discrete state models

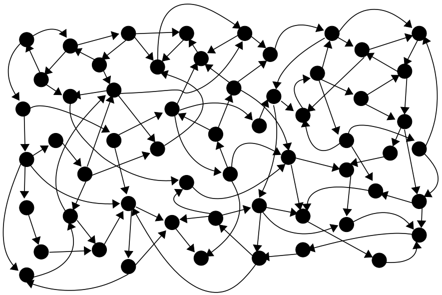


Under the SOS semantics a SPA model is mapped to a CTMC with global states determined by the local states of all the participating components.

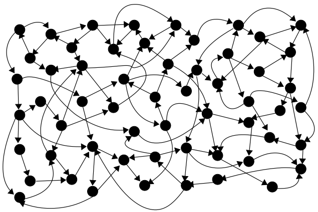
Solving discrete state models

Under the SOS semantics a SPA model is mapped to a CTMC with global states determined by the local states of all the participating components.



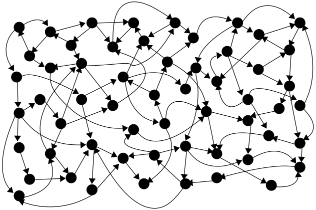


Solving discrete state models



When the size of the state space is not too large they are amenable to numerical solution (linear algebra) to determine a steady state or transient probability distribution.

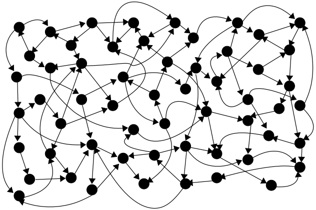
Solving discrete state models



When the size of the state space is not too large they are amenable to numerical solution (linear algebra) to determine a steady state or transient probability distribution.

$$Q = \begin{pmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,N} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,N} \\ \vdots & \vdots & & \vdots \\ q_{N,1} & q_{N,2} & \cdots & q_{N,N} \end{pmatrix}$$

Solving discrete state models



When the size of the state space is not too large they are amenable to numerical solution (linear algebra) to determine a steady state or transient probability distribution.

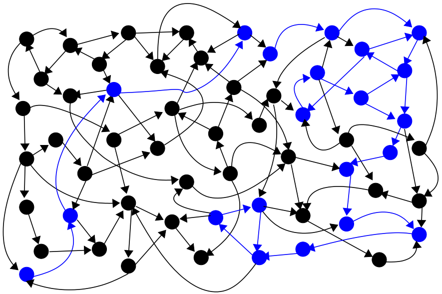
$$Q = \begin{pmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,N} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,N} \\ \vdots & \vdots & & \vdots \\ q_{N,1} & q_{N,2} & \cdots & q_{N,N} \end{pmatrix}$$

$$\pi(t) = (\pi_1(t), \pi_2(t), \ldots, \pi_N(t))$$

$$\pi(\infty)Q = 0$$

Solving discrete state models

Alternatively they may be studied using stochastic simulation. Each run generates a single trajectory through the state space. Many runs are needed in order to obtain average behaviours.



State space explosion

As the size of the state space becomes large it becomes infeasible to carry out numerical solution and extremely time-consuming to conduct stochastic simulation.

The Fluid Approximation Alternative

Fortunately there is an alternative: fluid approximation.

The Fluid Approximation Alternative

Fortunately there is an alternative: fluid approximation.

For a large class of models, just as the size of the state space becomes unmanageable, the models become amenable to an efficient, scale-free approximation.

The Fluid Approximation Alternative

Fortunately there is an alternative: fluid approximation.

For a large class of models, just as the size of the state space becomes unmanageable, the models become amenable to an efficient, scale-free approximation.

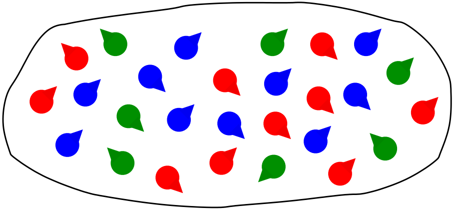
These are models which consist of populations.

Identity and Individuality

Population systems are constructed from many instances of a set of components.

Identity and Individuality

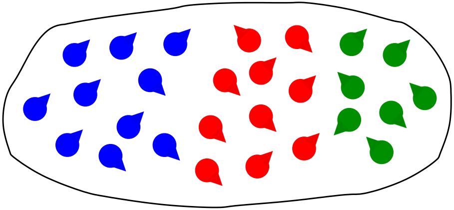
Population systems are constructed from many instances of a set of components.



If we cease to distinguish between instances of components we can form an aggregation or counting abstraction to reduce the state space.

Identity and Individuality

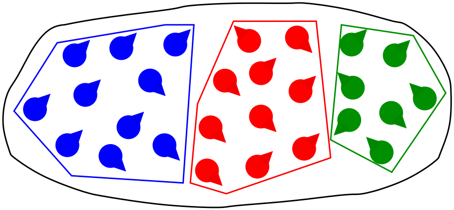
Population systems are constructed from many instances of a set of components.



If we cease to distinguish between instances of components we can form an aggregation or counting abstraction to reduce the state space.

Identity and Individuality

Population systems are constructed from many instances of a set of components.

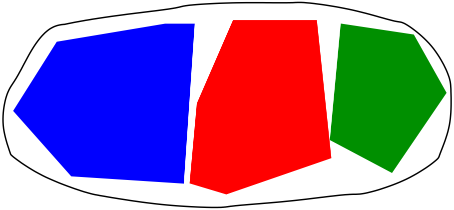


If we cease to distinguish between instances of components we can form an aggregation or counting abstraction to reduce the state space.

We may choose to disregard the identity of components.

Identity and Individuality

Population systems are constructed from many instances of a set of components.



If we cease to distinguish between instances of components we can form an aggregation or counting abstraction to reduce the state space.

We may choose to disregard the identity of components.

Even better reductions can be achieved when we no longer regard the components as individuals.

Population models

A shift in perspective allows us to model the interactions between individual components but then only consider the system as a whole as an interaction of populations.

Population models

A shift in perspective allows us to model the interactions between individual components but then only consider the system as a whole as an interaction of populations.

This allows us to model much larger systems than previously possible but in making the shift we are no longer able to collect any information about individuals in the system.

Population models

A shift in perspective allows us to model the interactions between individual components but then only consider the system as a whole as an interaction of populations.

This allows us to model much larger systems than previously possible but in making the shift we are no longer able to collect any information about individuals in the system.

To characterise the behaviour of a population we calculate the proportion of individuals within the population that are exhibiting certain behaviours rather than tracking individuals directly.

Population models

A shift in perspective allows us to model the interactions between individual components but then only consider the system as a whole as an interaction of populations.

This allows us to model much larger systems than previously possible but in making the shift we are no longer able to collect any information about individuals in the system.

To characterise the behaviour of a population we calculate the proportion of individuals within the population that are exhibiting certain behaviours rather than tracking individuals directly.

Furthermore we make a continuous approximation of how the proportions vary over time.

Continuous Approximation

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Continuous Approximation

Although in reality all state transitions are discrete, we can see that as the size of the population grows, the impact of each state change becomes smaller, and the error introduced by continuous approximation decreases.

Outline

Pragmatism and Expediency

Through pragmatism and expediency the representation of inherently discrete systems by collections of ordinary differential equations has been adopted in many areas of science, e.g. cell biology, ecology and epidemiology.

Pragmatism and Expediency

Through pragmatism and expediency the representation of inherently discrete systems by collections of ordinary differential equations has been adopted in many areas of science, e.g. cell biology, ecology and epidemiology.

ODEs may be derived through heuristics, or may be the accepted model (e.g. SIR models).

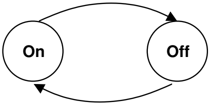
Pragmatism and Expediency

Through pragmatism and expediency the representation of inherently discrete systems by collections of ordinary differential equations has been adopted in many areas of science, e.g. cell biology, ecology and epidemiology.

ODEs may be derived through heuristics, or may be the accepted model (e.g. SIR models).

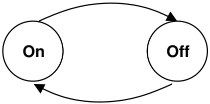
Nevertheless they may also be a fluid approximation which can be rigorously derived as the limit of a discrete model as the size of the population grows.

Population models — intuition



$Y(t)$

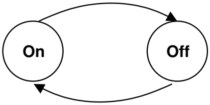
Population models — intuition

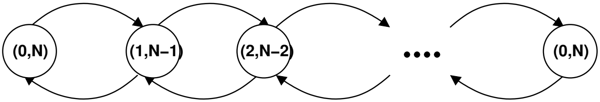


$Y(t)$

N copies: $Y_i^{(N)}$

Population models — intuition



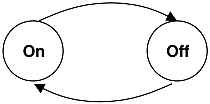


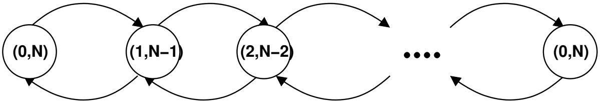
$Y(t)$

N copies: $Y_i^{(N)}$

$\mathbf{X}^{(N)}(t)$

Population models — intuition





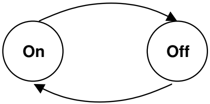
$Y(t)$

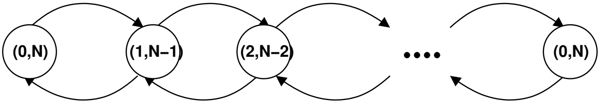
N copies: $Y_i^{(N)}$

$\mathbf{X}^{(N)}(t)$

$$X_j^{(N)} = \sum_{i=1}^{N} \mathbf{1}\{Y_i^{(N)} = j\}$$

Population models — intuition





$Y(t)$

N copies: $Y_i^{(N)}$

$\mathbf{X}^{(N)}(t)$

$$X_j^{(N)} = \sum_{i=1}^{N} \mathbf{1}\{Y_i^{(N)} = j\}$$

- $Y(t)$, $Y_i^{(N)}(t)$ and $\mathbf{X}^{(N)}(t)$ are all CTMCs;
- As N increases we get a sequence of CTMCs, $\mathbf{X}^{(N)}(t)$

Population state space

- The population process $\mathbf{X}^{(N)} = (X_1^{(N)}, \ldots, X_n^{(N)})$ has the dimension of the state space of $Y(t)$.

Population state space

- The population process $\mathbf{X}^{(N)} = (X_1^{(N)}, \ldots, X_n^{(N)})$ has the dimension of the state space of $Y(t)$.
- Importantly, its dimensions are independent of N.

Population state space

- The population process $\mathbf{X}^{(N)} = (X_1^{(N)}, \ldots, X_n^{(N)})$ has the dimension of the state space of $Y(t)$.
- Importantly, its dimensions are independent of N.
- Essentially we are making a counting abstraction and aggregation of the state space.

Population state space

- The population process $\mathbf{X}^{(N)} = (X_1^{(N)}, \ldots, X_n^{(N)})$ has the dimension of the state space of $Y(t)$.
- Importantly, its dimensions are independent of N.
- Essentially we are making a counting abstraction and aggregation of the state space.
- If we make the closed world assumption: $\sum_{j=1}^{n} X_j^{(N)} = N$

Population transitions

- The dynamics of the population models is expressed in terms of a set of possible transitions, $\mathcal{T}^{(N)}$.

Population transitions

- The dynamics of the population models is expressed in terms of a set of possible transitions, $\mathcal{T}^{(N)}$.
- Transitions are stochastic, and take an exponentially distributed time to happen.

Population transitions

- The dynamics of the population models is expressed in terms of a set of possible transitions, $\mathcal{T}^{(N)}$.
- Transitions are stochastic, and take an exponentially distributed time to happen.
- Their rate may depend on the current global state of the system.

Population transitions

- The dynamics of the population models is expressed in terms of a set of possible transitions, $\mathcal{T}^{(N)}$.
- Transitions are stochastic, and take an exponentially distributed time to happen.
- Their rate may depend on the current global state of the system.
- Each transition is specified by a rate function $r_{\tau}^{(N)}$, and by an update vector $\mathbf{v}_{\tau}$, specifying the impact of the event on the population vector.

Population transitions

- The dynamics of the population models is expressed in terms of a set of possible transitions, $\mathcal{T}^{(N)}$.
- Transitions are stochastic, and take an exponentially distributed time to happen.
- Their rate may depend on the current global state of the system.
- Each transition is specified by a rate function $r_\tau^{(N)}$, and by an update vector $\mathbf{v}_\tau$, specifying the impact of the event on the population vector.
- The infinitesimal generator matrix $Q^{(N)}$ of $\mathbf{X}^{(N)}(t)$ is defined as:

$$q_{\mathbf{x},\mathbf{x}'} = \sum \{ r_\tau(\mathbf{x}) \mid \tau \in \mathcal{T},\ \mathbf{x}' = \mathbf{x} + \mathbf{v}_\tau \}.$$

Population models — summary of notation

Individuals

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

Population models — summary of notation

Individuals

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

System variables

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

Population models — summary of 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 $\tau \in \mathcal{T}$:

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

Scaling Conditions

Scaling assumptions

- We have a sequence $\mathbf{X}^{(N)}$ of population CTMCs.
- We normalise such models, dividing variables by N:

$$\hat{\mathbf{X}} = \frac{\mathbf{X}}{N}$$

Scaling Conditions

Scaling assumptions

- We have a sequence $\mathbf{X}^{(N)}$ of population CTMCs.
- We normalise such models, dividing variables by N:

$$\hat{\mathbf{X}} = \frac{\mathbf{X}}{N}$$

occupancy measures

Scaling Conditions

Scaling assumptions

- We have a sequence $\mathbf{X}^{(N)}$ of population CTMCs.
- We normalise such models, dividing variables by N:

$$\hat{\mathbf{X}} = \frac{\mathbf{X}}{N} \qquad \text{occupancy measures}$$

- for each $\tau \in \mathcal{T}^{(N)}$
  - the normalised update is $\hat{\mathbf{v}} = \mathbf{v}/N$
  - there is a normalised rate function $\hat{r}_\tau(\hat{\mathbf{X}})$

Scaling Conditions

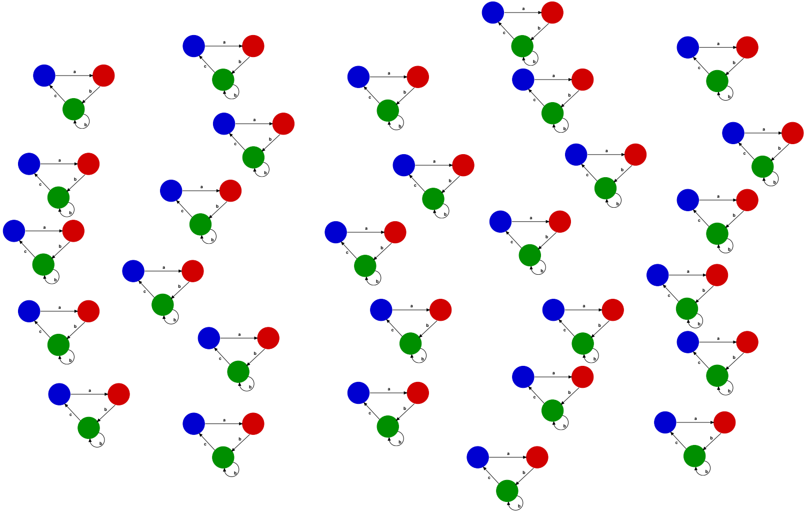
Scaling assumptions

- We have a sequence $\mathbf{X}^{(N)}$ of population CTMCs.
- We normalise such models, dividing variables by N:

$$\hat{\mathbf{X}} = \frac{\mathbf{X}}{N} \qquad \text{occupancy measures}$$

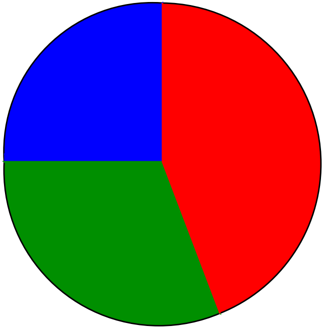
- for each $\tau \in \mathcal{T}^{(N)}$
  - the normalised update is $\hat{\mathbf{v}} = \mathbf{v}/N$
  - there is a normalised rate function $\hat{r}_\tau(\hat{\mathbf{X}})$
- $\forall \tau$ assume there exists a bounded and Lipschitz continuous function $f_\tau(\hat{\mathbf{X}})$, the limit rate function on normalised variables, independent of N, such that $\frac{1}{N}\hat{r}_\tau^{(N)}(\mathbf{x}) \to f_\tau(\mathbf{x})$ uniformly.

Normalised process — intuition



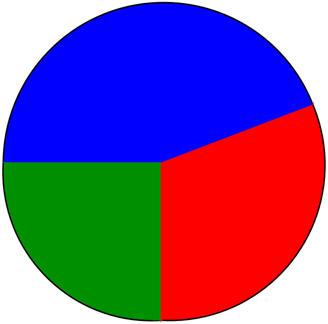
Normalised process — intuition

The whole population is represented as a single process.



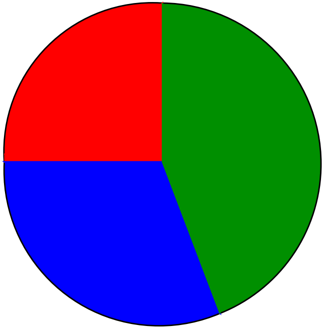
Normalised process — intuition

The whole population is represented as a single process.



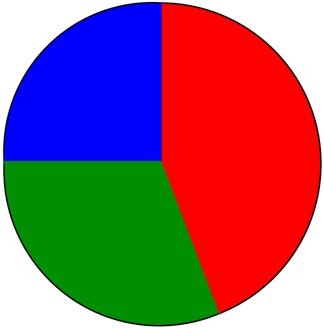
Normalised process — intuition

The whole population is represented as a single process.



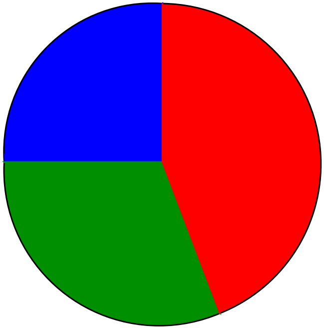
Normalised process — intuition

The whole population is represented as a single process.



Normalised process — intuition

The whole population is represented as a single process.



Even when the number of individuals varies ($N \longrightarrow \infty$) the processes remain comparable.

Drift

Drift

The drift $F^{(N)}(\hat{\mathbf{X}})$ — the mean instantaneous increment of model variables in state $\hat{\mathbf{X}}$ — is defined as

$$F^{(N)}(\hat{\mathbf{X}}) = \sum_{\tau \in \hat{\mathcal{T}}} \frac{1}{N} \mathbf{v}_\tau \, \hat{r}_\tau^{(N)}(\hat{\mathbf{X}})$$

Drift

Drift

The drift $F^{(N)}(\hat{\mathbf{X}})$ — the mean instantaneous increment of model variables in state $\hat{\mathbf{X}}$ — is defined as

$$F^{(N)}(\hat{\mathbf{X}}) = \sum_{\tau \in \hat{\mathcal{T}}} \frac{1}{N} \mathbf{v}_\tau \, \hat{r}_\tau^{(N)}(\hat{\mathbf{X}})$$

Limit Drift

Let f_τ be the limit rate functions.

The limit drift of the model $\hat{\mathcal{X}}^{(N)}$ is

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

and $F^{(N)}(\mathbf{x}) \to F(\mathbf{x})$ uniformly as $N \longrightarrow \infty$.

Fluid ODE and Fluid approximation theorem

Fluid ODE

The fluid ODE is

$$\frac{d\mathbf{x}}{dt} = F(\mathbf{x}), \quad with\ \mathbf{x}(0) = \mathbf{x_0} \in S.$$

Since F is Lipschitz (all f_τ are), this ODE has a unique solution $\mathbf{x}(t)$ starting from $\mathbf{x_0}$.

Deterministic Approximation Theorem (Kurtz)

Assume that $\exists\, \mathbf{x_0} \in S$ such that $\hat{\mathbf{X}}^{(N)}(0) \to \mathbf{x_0}$ in probability. Then, for any finite time horizon $T < \infty$, it holds that as $N \longrightarrow \infty$:

$$\mathbb{P}\left\{ \sup_{0 \leq t \leq T} ||\hat{\mathbf{X}}^{(N)}(t) - \mathbf{x}(t)|| > \varepsilon \right\} \to 0.$$

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

Fluid Approximation ODEs

The fluid approximation ODEs can be interpreted in two different ways:

- as an approximation of the average of the system (usually a first order approximation). This is often termed a mean field approximation.

Fluid Approximation ODEs

The fluid approximation ODEs can be interpreted in two different ways:

- as an approximation of the average of the system (usually a first order approximation). This is often termed a mean field approximation.
- as an approximate description of system trajectories for large populations.

Fluid Approximation ODEs

The fluid approximation ODEs can be interpreted in two different ways:

- as an approximation of the average of the system (usually a first order approximation). This is often termed a mean field approximation.
- as an approximate description of system trajectories for large populations.

We focus on the second interpretation — a functional version of the Law of Large Numbers.

Fluid Approximation ODEs

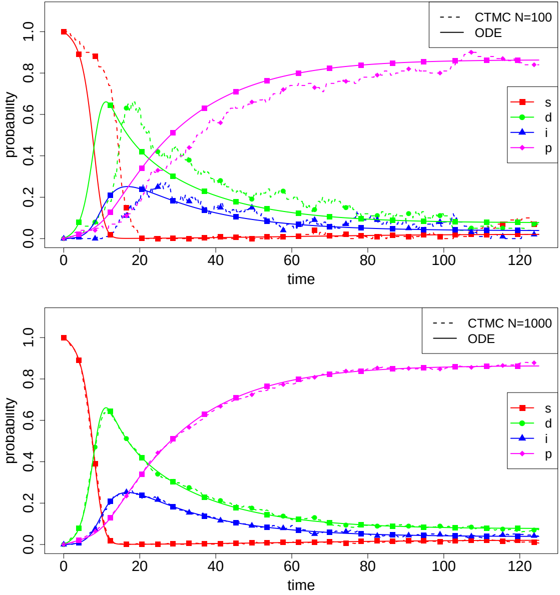
The fluid approximation ODEs can be interpreted in two different ways:

- as an approximation of the average of the system (usually a first order approximation). This is often termed a mean field approximation.
- as an approximate description of system trajectories for large populations.

We focus on the second interpretation — a functional version of the Law of Large Numbers.

Instead of having a sequence of random variables, converging to a deterministic value, here we have a sequence of CTMCs for increasing population size, which converge to a deterministic trajectory, the solution of the fluid ODE.

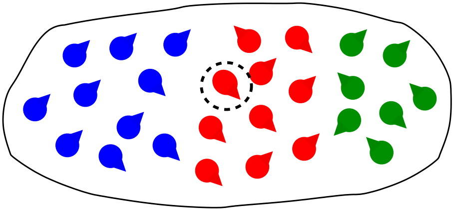
Illustrative trajectories



Comparison of the limit fluid ODE and a single stochastic trajectory of a network epidemic example, for total populations $N = 100$ and $N = 1000$.

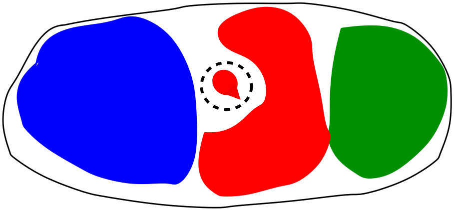
Implications of the Deterministic Approximation Theorem

The Theorem implies that in the limit the dynamics of a single agent becomes independent of other agents — it will sense them only through the collective system state, or mean field, described by the fluid limit.



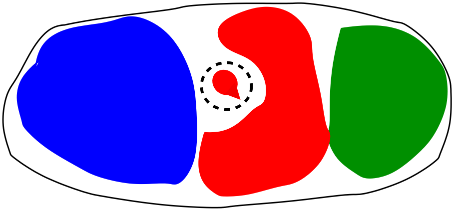
Implications of the Deterministic Approximation Theorem

The Theorem implies that in the limit the dynamics of a single agent becomes independent of other agents — it will sense them only through the collective system state, or mean field, described by the fluid limit.



Implications of the Deterministic Approximation Theorem

The Theorem implies that in the limit the dynamics of a single agent becomes independent of other agents — it will sense them only through the collective system state, or mean field, described by the fluid limit.



This asymptotic decoupling allows us to find a simple, time-inhomogenous, Markov chain for the evolution of the single agent, a result often known as fast simulation.

Focusing on one individual

- We focus on a single individual $Y_h^{(N)}(t)$, a (Markov) process on the state space $S = \{1, \ldots, n\}$, conditional on the global state of the complete population $\hat{\mathbf{X}}^{(N)}(t)$.

Focusing on one individual

- We focus on a single individual $Y_h^{(N)}(t)$, a (Markov) process on the state space $S = \{1, \ldots, n\}$, conditional on the global state of the complete population $\hat{\mathbf{X}}^{(N)}(t)$.
- Rate $q_{ij}(\hat{\mathbf{X}})$ depends on the global system state and $\hat{\mathbf{X}}^{(N)}(t)$.

Focusing on one individual

- We focus on a single individual $Y_h^{(N)}(t)$, a (Markov) process on the state space $S = \{1, \ldots, n\}$, conditional on the global state of the complete population $\hat{\mathbf{X}}^{(N)}(t)$.
- Rate $q_{ij}(\hat{\mathbf{X}})$ depends on the global system state and $\hat{\mathbf{X}}^{(N)}(t)$.
- Its transition rates q_{ij}, are obtained projecting on a single agent the rate of global transitions that induce a change of state of at least one agent from i to j.

Focusing on one individual

- We focus on a single individual $Y_h^{(N)}(t)$, a (Markov) process on the state space $S = \{1, \ldots, n\}$, conditional on the global state of the complete population $\hat{\mathbf{X}}^{(N)}(t)$.
- Rate $q_{ij}(\hat{\mathbf{X}})$ depends on the global system state and $\hat{\mathbf{X}}^{(N)}(t)$.
- Its transition rates q_{ij}, are obtained projecting on a single agent the rate of global transitions that induce a change of state of at least one agent from i to j.
- However, by the theorem, as $N \longrightarrow \infty$, the stochastic fluctuations of $\hat{\mathbf{X}}^{(N)}(t)$ tend to vanish, and the stochastic behaviour of $Y_h^{(N)}(t)$ can be approximated by making it dependent only on the fluid limit $\mathbf{x}(t)$.

Focusing on one individual

- We focus on a single individual $Y_h^{(N)}(t)$, a (Markov) process on the state space $S = \{1, \ldots, n\}$, conditional on the global state of the complete population $\hat{\mathbf{X}}^{(N)}(t)$.
- Rate $q_{ij}(\hat{\mathbf{X}})$ depends on the global system state and $\hat{\mathbf{X}}^{(N)}(t)$.
- Its transition rates q_{ij}, are obtained projecting on a single agent the rate of global transitions that induce a change of state of at least one agent from i to j.
- However, by the theorem, as $N \longrightarrow \infty$, the stochastic fluctuations of $\hat{\mathbf{X}}^{(N)}(t)$ tend to vanish, and the stochastic behaviour of $Y_h^{(N)}(t)$ can be approximated by making it dependent only on the fluid limit $\mathbf{x}(t)$.
- Thus we construct the time-inhomogeneous CTMC $z(t)$ with state space S and rates $q_{ij}(\mathbf{x}(t))$.

Fast Simulation

Fast Simulation Theorem (Darling and Norris)

For any finite time horizon $T < \infty$,

$$\mathbb{P}\{Y_h^{(N)}(t) \neq z(t), \text{ for some } t \leq T\} \to 0, \text{ as } N \to \infty.$$

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

Fast Simulation

Fast Simulation Theorem (Darling and Norris)

For any finite time horizon $T < \infty$,

$$\mathbb{P}\{Y_h^{(N)}(t) \neq z(t), \text{ for some } t \leq T\} \to 0, \text{ as } N \to \infty.$$

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

This theorem states that, in the limit of an infinite population, each agent will behave independently from all the others, sensing only the mean state of the global system, described by the fluid limit $\mathbf{x}(t)$.

Outline

Process Algebra for Population Systems

Process algebra are well-suited for constructing models of population systems:

- Developed to represent concurrent behaviour compositionally;

Process Algebra for Population Systems

Process algebra are well-suited for constructing models of population systems:

- Developed to represent concurrent behaviour compositionally;
- Represent the interactions between individuals explicitly;

Process Algebra for Population Systems

Process algebra are well-suited for constructing models of population systems:

- Developed to represent concurrent behaviour compositionally;
- Represent the interactions between individuals explicitly;
- Populations are readily and rigorously identified;

Process Algebra for Population Systems

Process algebra are well-suited for constructing models of population systems:

- Developed to represent concurrent behaviour compositionally;
- Represent the interactions between individuals explicitly;
- Populations are readily and rigorously identified;
- Stochastic extensions allow the dynamics of system behaviour to be captured;

Process Algebra for Population Systems

Process algebra are well-suited for constructing models of population systems:

- Developed to represent concurrent behaviour compositionally;
- Represent the interactions between individuals explicitly;
- Populations are readily and rigorously identified;
- Stochastic extensions allow the dynamics of system behaviour to be captured;
- Incorporate formal apparatus for reasoning about the behaviour of systems through model checking.

Process Algebra for Population Systems

Process algebra are well-suited for constructing models of population systems:

- Developed to represent concurrent behaviour compositionally;
- Represent the interactions between individuals explicitly;
- Populations are readily and rigorously identified;
- Stochastic extensions allow the dynamics of system behaviour to be captured;
- Incorporate formal apparatus for reasoning about the behaviour of systems through model checking.

The major impediment is state space explosion and fluid approximation offers a solution to that problem.

Fluid semantics for Stochastic Process Algebras

- Incorporating fluid approximation into a formal high-level language used for constructing CTMC models offers quantitative scalable analysis which is immune to state space explosion.

Fluid semantics for Stochastic Process Algebras

- Incorporating fluid approximation into a formal high-level language used for constructing CTMC models offers quantitative scalable analysis which is immune to state space explosion.
- Indeed, accuracy increases as the size of the model grows.

Fluid semantics for Stochastic Process Algebras

- Incorporating fluid approximation into a formal high-level language used for constructing CTMC models offers quantitative scalable analysis which is immune to state space explosion.
- Indeed, accuracy increases as the size of the model grows.
- Embedding the approach in a formal language offers the possibility to establish the conditions for convergence at the language level via the semantics,

Fluid semantics for Stochastic Process Algebras

- Incorporating fluid approximation into a formal high-level language used for constructing CTMC models offers quantitative scalable analysis which is immune to state space explosion.
- Indeed, accuracy increases as the size of the model grows.
- Embedding the approach in a formal language offers the possibility to establish the conditions for convergence at the language level via the semantics,
- This removes the requirement to fulfil the proof obligation on a model-by-model basis.

Fluid semantics for Stochastic Process Algebras

- Incorporating fluid approximation into a formal high-level language used for constructing CTMC models offers quantitative scalable analysis which is immune to state space explosion.
- Indeed, accuracy increases as the size of the model grows.
- Embedding the approach in a formal language offers the possibility to establish the conditions for convergence at the language level via the semantics,
- This removes the requirement to fulfil the proof obligation on a model-by-model basis.
- Moreover the derivation of the ODEs can be automated in the implementation of the language.

Multiple agents

Kurtz's Theorem is based on the notion of a single agent class — many instances of one sequential component.

Multiple agents

Kurtz's Theorem is based on the notion of a single agent class — many instances of one sequential component.

But in a process algebra model we typically work with multiple components composed to evolve concurrently.

Multiple agents

Kurtz's Theorem is based on the notion of a single agent class — many instances of one sequential component.

But in a process algebra model we typically work with multiple components composed to evolve concurrently.

We construct a single agent class in the population CTMC but partition the state space S into subsets, each of which represents the states of a distinct component, and such that there are no transitions between subsets.

Multiple agents

Kurtz's Theorem is based on the notion of a single agent class — many instances of one sequential component.

But in a process algebra model we typically work with multiple components composed to evolve concurrently.

We construct a single agent class in the population CTMC but partition the state space S into subsets, each of which represents the states of a distinct component, and such that there are no transitions between subsets.

The agents whose initial state is in each subset correspond to that component.

Deriving a Fluid Approximation of a SPA model

The aim is to represent the CTMC implicitly (avoiding state space explosion), and to generate the set of ODEs which are the fluid limit of that CTMC.

Deriving a Fluid Approximation of a SPA model

The aim is to represent the CTMC implicitly (avoiding state space explosion), and to generate the set of ODEs which are the fluid limit of that CTMC.

The existing SOS semantics is not suitable for this purpose because it constructs the state space of the CTMC explicitly.

Deriving a Fluid Approximation of a SPA model

The aim is to represent the CTMC implicitly (avoiding state space explosion), and to generate the set of ODEs which are the fluid limit of that CTMC.

The existing SOS semantics is not suitable for this purpose because it constructs the state space of the CTMC explicitly.

SPA MODEL —SOS rules→ LABELLED TRANSITION SYSTEM —state transition diagram→ CTMC **Q**

Deriving a Fluid Approximation of a SPA model

The aim is to represent the CTMC implicitly (avoiding state space explosion), and to generate the set of ODEs which are the fluid limit of that CTMC.

The existing SOS semantics is not suitable for this purpose because it constructs the state space of the CTMC explicitly.

We define a structured operational semantics which defines the possible transitions of an arbitrary abstract state and from this derive the ODEs.

Deriving a Fluid Approximation of a SPA model

The aim is to represent the CTMC implicitly (avoiding state space explosion), and to generate the set of ODEs which are the fluid limit of that CTMC.

The existing SOS semantics is not suitable for this purpose because it constructs the state space of the CTMC explicitly.

We define a structured operational semantics which defines the possible transitions of an arbitrary abstract state and from this derive the ODEs.

Deriving a Fluid Approximation of a SPA model

The aim is to represent the CTMC implicitly (avoiding state space explosion), and to generate the set of ODEs which are the fluid limit of that CTMC.

The existing SOS semantics is not suitable for this purpose because it constructs the state space of the CTMC explicitly.

We define a structured operational semantics which defines the possible transitions of an arbitrary abstract state and from this derive the ODEs.

Similar work has been done for WSCCS, sCCP, Stochastic CCS, Kappa, Bio-PEPA and Grouped PEPA.

Fluid Structured Operational Semantics

In order to get to the implicit representation of the CTMC we need to:

Fluid Structured Operational Semantics

In order to get to the implicit representation of the CTMC we need to:

1. Remove excess components to identify the counting abstraction of the process (Context Reduction)

Fluid Structured Operational Semantics

In order to get to the implicit representation of the CTMC we need to:

1. Remove excess components to identify the counting abstraction of the process (Context Reduction)
2. Collect the transitions of the reduced context as symbolic updates on the state representation (Jump Multiset)

Fluid Structured Operational Semantics

In order to get to the implicit representation of the CTMC we need to:

1. Remove excess components to identify the counting abstraction of the process (Context Reduction)
2. Collect the transitions of the reduced context as symbolic updates on the state representation (Jump Multiset)
3. Calculate the rate of the transitions in terms of an arbitrary state of the CTMC.

Fluid Structured Operational Semantics

In order to get to the implicit representation of the CTMC we need to:

1. Remove excess components to identify the counting abstraction of the process (Context Reduction)
2. Collect the transitions of the reduced context as symbolic updates on the state representation (Jump Multiset)
3. Calculate the rate of the transitions in terms of an arbitrary state of the CTMC.

Once this is done we can extract the vector field $F_{\mathcal{M}}(x)$ from the jump multiset, under the assumption that the population size tends to infinity.

M.Tribastone, S.Gilmore and J.Hillston. Scalable Differential Analysis of Process Algebra Models. IEEE TSE 2012.

Consistency results

- The vector field $\mathcal{F}(x)$ is Lipschitz continuous i.e. all the rate functions governing transitions in the process algebra satisfy local continuity conditions.

Consistency results

- The vector field $\mathcal{F}(x)$ is Lipschitz continuous i.e. all the rate functions governing transitions in the process algebra satisfy local continuity conditions.
- Thus the hypotheses of the Deterministic Approximation Theorem are satisfied.

Consistency results

- The vector field $\mathcal{F}(x)$ is Lipschitz continuous i.e. all the rate functions governing transitions in the process algebra satisfy local continuity conditions.
- Thus the hypotheses of the Deterministic Approximation Theorem are satisfied.
- The generated ODEs are the fluid limit of the family of CTMCs and so approximate the discrete behaviour as the size of the system grows.

Consistency results

- The vector field $\mathcal{F}(x)$ is Lipschitz continuous i.e. all the rate functions governing transitions in the process algebra satisfy local continuity conditions.
- Thus the hypotheses of the Deterministic Approximation Theorem are satisfied.
- The generated ODEs are the fluid limit of the family of CTMCs and so approximate the discrete behaviour as the size of the system grows.
- Moreover Lipschitz continuity of the vector field guarantees existence and uniqueness of the solution to the initial value problem.

M.Tribastone, S.Gilmore and J.Hillston. Scalable Differential Analysis of Process Algebra Models. IEEE TSE 2012.

Quantitative properties

The derived vector field $\mathcal{F}(x)$, gives an approximation of the expected count for each population over time.

Quantitative properties

The derived vector field $\mathcal{F}(x)$, gives an approximation of the expected count for each population over time.

This has been extended in a number of ways:

- Fluid rewards which can be safely calculated from the fluid expectation trajectories.

M.Tribastone, J.Ding, S.Gilmore and J.Hillston. Fluid Rewards for a Stochastic Process Algebra. IEEE TSE 2012.

Quantitative properties

The derived vector field $\mathcal{F}(x)$, gives an approximation of the expected count for each population over time.

This has been extended in a number of ways:

- Fluid rewards which can be safely calculated from the fluid expectation trajectories.

M.Tribastone, J.Ding, S.Gilmore and J.Hillston. Fluid Rewards for a Stochastic Process Algebra. IEEE TSE 2012.

- Vector fields have been defined to approximate higher moments.

R.A.Hayden and J.T.Bradley. A fluid analysis framework for a Markovian process algebra. TCS 2010.

Quantitative properties

The derived vector field $\mathcal{F}(x)$, gives an approximation of the expected count for each population over time.

This has been extended in a number of ways:

- Fluid rewards which can be safely calculated from the fluid expectation trajectories.

M.Tribastone, J.Ding, S.Gilmore and J.Hillston. Fluid Rewards for a Stochastic Process Algebra. IEEE TSE 2012.

- Vector fields have been defined to approximate higher moments.

R.A.Hayden and J.T.Bradley. A fluid analysis framework for a Markovian process algebra. TCS 2010.

- Fluid approximation of passage times have been defined.

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

Outline

Fluid model checking

Since the vector field records only deterministic behaviour, LTL model checking can be used over a trace to give boolean results. But for the systems we are interested in we would like some more quantified answers, in the style of stochastic model checking.

Fluid model checking

Since the vector field records only deterministic behaviour, LTL model checking can be used over a trace to give boolean results. But for the systems we are interested in we would like some more quantified answers, in the style of stochastic model checking.

Work on this is on-going but there are initial results for:

- CSL properties of a single agent within a population.

  L.Bortolussi and J.Hillston. Fluid model checking. CONCUR 2012.

- The fraction of a population that satisfies a property expressed as a one-clock deterministic timed automaton.

  L.Bortolussi and R.Lanciani. Central Limit Approximation for Stochastic Model Checking. QEST 2013.

- Model checking for PCTL single agent properties in discrete-time, synchronous clock population processes.

  D.Latella, M.Loreti and M.Massink. On-the-fly Fast Mean-Field Model-Checking. TGC 2013.

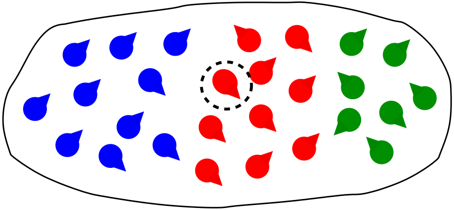
CSL model checking of a single agent

We consider properties of a single agent within a population, expressed in the Continuous Stochastic Logic (CSL), usually used for model checking CTMCs, and exploit fast simulation.

CSL model checking of a single agent

We consider properties of a single agent within a population, expressed in the Continuous Stochastic Logic (CSL), usually used for model checking CTMCs, and exploit fast simulation.

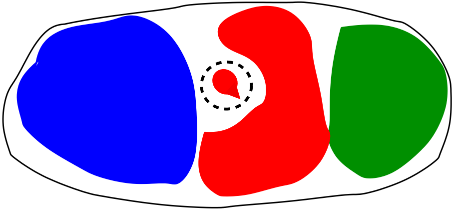
We consider an arbitrary member of the population.



CSL model checking of a single agent

We consider properties of a single agent within a population, expressed in the Continuous Stochastic Logic (CSL), usually used for model checking CTMCs, and exploit fast simulation.

We consider an arbitrary member of the population.

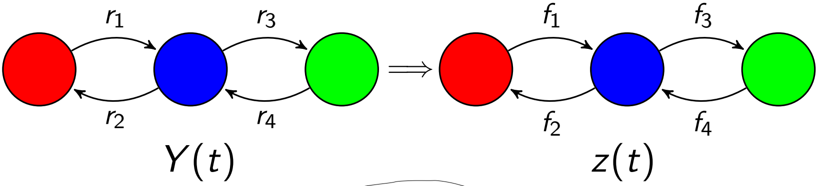


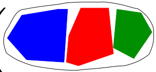
This agent is kept discrete, making transitions between its discrete states, but all other agents are treated as a mean-field influencing the behaviour of this agent.

Inhomogeneous CTMC

The transition rates within the discrete-event representation will depend on the rest of the population.

i.e. it will depend on the vector field capturing the behaviour of the residual population.

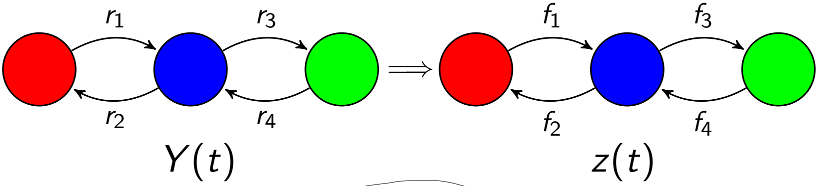


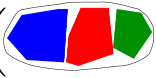
where $f_i = f\Big($  $\Big)$

Inhomogeneous CTMC

The transition rates within the discrete-event representation will depend on the rest of the population.

i.e. it will depend on the vector field capturing the behaviour of the residual population.



where $f_i = f\Big($  $\Big)$

It is an inhomogeneous CTMC, with rates that vary with time according to the mean field.

CSL model checking for CTMC

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

$$\mathcal{P}_{\bowtie p}\left(\varphi_1 \, U^{[0,T]} \, \varphi_2\right)$$

Time-homogeneous CTMC

We check this formula by computing, for each state $s \in S$, the probability of paths satisfying $\varphi_1 \, U^{[0,T]} \, \varphi_2$ and then comparing this probability $\bowtie p$.

This is done via transient analysis on the chain in which $\neg\varphi_1$ and φ_2 states are made absorbing.

CSL model checking for CTMC

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

$$\mathcal{P}_{\bowtie p}\left(\varphi_1 \, U^{[0,T]} \, \varphi_2\right)$$

Time-homogeneous CTMC

We check this formula by computing, for each state $s \in S$, the probability of paths satisfying $\varphi_1 \, U^{[0,T]} \, \varphi_2$ and then comparing this probability $\bowtie p$.

This is done via transient analysis on the chain in which $\neg\varphi_1$ 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 ICTMC

Again consider a CTMC with state space S and rates given by $Q = Q(t)$ and the formula $\mathcal{P}_{\bowtie p}(\varphi_1 \, U^{[0,T]} \, \varphi_2)$.

CSL model checking for ICTMC

Again consider a CTMC with state space S and rates given by $Q = Q(t)$ and the formula $\mathcal{P}_{\bowtie p}(\varphi_1 \, U^{[0,T]} \, \varphi_2)$.

Time-inhomogeneous CTMC

We check this formula by computing, for each state $s \in S$, the probability of paths satisfying $\varphi_1 \, U^{[0,T]} \, \varphi_2$ and then comparing this probability $\bowtie p$.

Again this is done via transient analysis, based on the Kolmogorov equations, in which $\neg\varphi_1$ and φ_2 states are made absorbing.

CSL model checking for ICTMC

Again consider a CTMC with state space S and rates given by $Q = Q(t)$ and the formula $\mathcal{P}_{\bowtie p}(\varphi_1\, U^{[0,T]}\, \varphi_2)$.

Time-inhomogeneous CTMC

We check this formula by computing, for each state $s \in S$, the probability of paths satisfying $\varphi_1\, U^{[0,T]}\, \varphi_2$ and then comparing this probability $\bowtie p$.

Again this is done via transient analysis, based on the Kolmogorov equations, in which $\neg\varphi_1$ and φ_2 states are made absorbing.

But:

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.

Time-dependent truth

- When computing the truth value of an until formula, we obtain a time dependent value $\textbf{true}(\varphi, s, t)$ in each state.

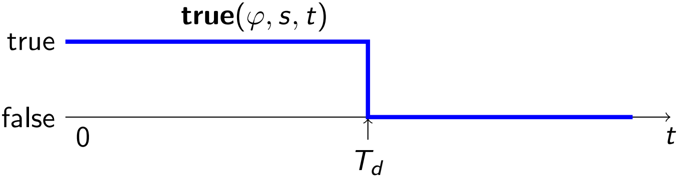
Time-dependent truth

- When computing the truth value of an until formula, we obtain a time dependent value $\mathbf{true}(\varphi, s, t)$ in each state.
- When we consider nested temporal operators, we need to take this into account.

Time-dependent truth

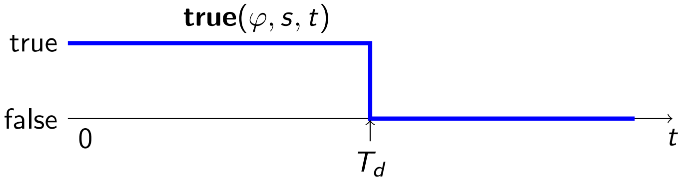
- When computing the truth value of an until formula, we obtain a time dependent value **true**(φ, s, t) in each state.
- When we consider nested temporal operators, we need to take this into account.
- The problem is that in this case the topology of goal and unsafe states in the CTMC can change in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

Time dependent truth



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

Time dependent truth



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

Fortunately

Discontinuities happen at specific and fixed time instants.

We can carry out the transient solution, using Kolmogorov equations, piecewise.

At each discontinuity event, we also have to appropriately change the absorbing structure of the Q matrix.

Convergence of CSL truth

- Consider convergence of CSL properties: will properties that are true in z_k eventually be true in $Y_k^{(N)}$?

Asymptotic Correctness Theorem

Let $\varphi = \varphi(\mathbf{p})$ 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 $\forall N \geq N_0$

$$s, 0 \models_{Y_k^{(N)}} \varphi \Leftrightarrow s, 0 \models_{z_k} \varphi.$$

L.Bortolussi and J.Hillston. Fluid model checking. CONCUR 2012.

Outline

Future Perspectives

Some limitations which I have included in this talk have already been addressed:

- The closed world assumption — the theory can be generalised to apply to growing or declining populations of agents.
- Finite time horizon — for models which have ergodic behaviour, the results can be extended to consider infinite time horizons.

Future Perspectives

Some limitations which I have included in this talk have already been addressed:

- The closed world assumption — the theory can be generalised to apply to growing or declining populations of agents.
- Finite time horizon — for models which have ergodic behaviour, the results can be extended to consider infinite time horizons.

On-going work is seeking to incorporate infinite time horizons into fluid model checking to allow consideration of unbounded until formulae.

Future Perspectives

Some limitations which I have included in this talk have already been addressed:

- The closed world assumption — the theory can be generalised to apply to growing or declining populations of agents.
- Finite time horizon — for models which have ergodic behaviour, the results can be extended to consider infinite time horizons.

On-going work is seeking to incorporate infinite time horizons into fluid model checking to allow consideration of unbounded until formulae.

We also aim to extend the result to consider global properties of the system, in addition to those focussed on individual agents.

Thanks

Thanks

Thanks to the other members of the QUANTICOL project

quanticol

www.quanticol.eu