Contextual Lumpability

Jane Hillston

School of Informatics
The University of Edinburgh
Scotland

joint work with Carla Piazza (Udine), Andrea Marin and Sabina Rossi (Ca’ Foscari),
Overview

- **Lumbability** (Kemeny and Snell, 1960) has long been established as an exact aggregation techniques for Markov chains.
Overview

- **Lumpability** (Kemeny and Snell, 1960) has long been established as an exact aggregation technique for Markov chains.

- In the context of stochastic process algebras, **Markovian bisimulations**, have been shown to characterise lumpability in the sense that partitioning the state space on the basis of such an equivalence relation generates a lumpable partition of the underlying state space.
Overview

- **Lumpability** (Kemeny and Snell, 1960) has long been established as an exact aggregation techniques for Markov chains.

- In the context of stochastic process algebras, **Markovian bisimulations**, have been shown to characterise lumpability in the sense that partitioning the state space on the basis of such an equivalence relation generates a lumpable partition of the underlying state space.

- In the stochastic process algebra PEPA, this equivalence relation is termed **strong equivalence**.
Overview

- In this paper we introduced a relaxed form of lumpability for PEPA models, termed contextual lumpability.
Overview

In this paper we introduced a relaxed form of lumpability for PEPA models, termed contextual lumpability.

We also define a new equivalence relation for PEPA models, termed lumpable bisimilarity.
Overview

- In this paper we introduced a relaxed form of lumpability for PEPA models, termed **contextual lumpability**.

- We also define a new equivalence relation for PEPA models, termed **lumpable bisimilarity**.

- We show that lumpable bisimilarity is a **characterisation** of contextual lumpability and induces the largest contextual lumping of the underlying Markov chain.
Overview

- In this paper we introduced a relaxed form of lumpability for PEPA models, termed **contextual lumpability**.

- We also define a new equivalence relation for PEPA models, termed **lumpable bisimilarity**.

- We show that lumpable bisimilarity is a **characterisation** of contextual lumpability and induces the largest contextual lumping of the underlying Markov chain.

- Finally we provide an **algorithm** for lumpable bisimilarity.
Outline

1. Lumpability
2. Strong Equivalence in PEPA
   - Example
3. Contextual Lumpability
4. Conclusions
Equivalence relations and state space explosion

It is well-known that Markovian based modelling techniques are prone to state space explosion.
Equivalence relations and state space explosion

It is well-known that Markovian based modelling techniques are prone to state space explosion.

Model aggregation: use a state-state equivalence to establish a partition of the state space of a model, and replace each set of states by one macro-state.
Equivalence relations and state space explosion

It is well-known that Markovian based modelling techniques are prone to state space explosion.

Model aggregation: use a state-state equivalence to establish a partition of the state space of a model, and replace each set of states by one macro-state.
Equivalence relations and state space explosion

It is well-known that Markovian based modelling techniques are prone to state space explosion.

Model aggregation: use a state-state equivalence to establish a partition of the state space of a model, and replace each set of states by one macro-state.

Models based on model aggregation/state-state equivalence will not in general be exact, because the macro states will not in general preserve the Markov property.
Equivalence relations and state space explosion

It is well-known that Markovian based modelling techniques are prone to state space explosion.

Model aggregation: use a state-state equivalence to establish a partition of the state space of a model, and replace each set of states by one macro-state.

Models based on model aggregation/state-state equivalence will not in general be exact, because the macro states will not in general preserve the Markov property.

But it has long been established that if the aggregation is based on a lumpable partition then the Markov property is preserved [Kemeny and Snell, 1960].
Reducing by lumpability
Reducing by lumpability
Reducing by lumpability
Lumpability

A CTMC, \( \{X_i\} \), is (ordinarily) lumpable with respect to some partition \( \chi = \{X_{[k]}\} \) if for any \( X_{[k]}, X_{[l]} \in \chi \) with \( k \neq l \) and \( X_i, X_j \in X_{[k]} \),

\[
q(X_i, X_{[l]}) = q(X_j, X_{[l]})
\]

where \( q(X_i, X_{[l]}) \) is the aggregated transition rate from \( X_i \) to all states in \( X_{[l]} \), i.e., \( q(X_i, X_{[l]}) = \sum_{X_m \in X_{[l]}} q(X_i, X_m) \).
Lumpable Relation

Lumpable relation

A relation \( R \subseteq C \times C \) over PEPA components is **lumpable** if for any component \( P \), \( ds(P)/R \) induces a lumpable partition on the state space of the CTMC corresponding to \( P \).
Lumpable Relation

Lumpable relation
A relation \( R \subseteq C \times C \) over PEPA components is **lumpable** if for any component \( P \), \( ds(P)/R \) induces a lumpable partition on the state space of the CTMC corresponding to \( P \).

Proposition
Let \( I \) be a set of indices and \( R_i \) be a lumpable relation for all \( i \in I \). Then the union, \( R = \bigcup_{i \in I} R_i \), is also a lumpable relation.
Models are constructed from components which engage in activities.

\[ (\alpha, r).P \]

- action type or name
- activity rate (parameter of an exponential distribution)
- component/derivative
Models are constructed from components which engage in activities.

\[(\alpha, r).P\]

- action type or name
- activity rate (parameter of an exponential distribution)
- component/derivative

The language is used to generate a CTMC for performance modelling.
Stochastic Process Algebra

- Models are constructed from *components* which engage in activities.

\[(\alpha, r).P\]

- 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.

\[
(a, r).P
\]

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

SPA MODEL \rightarrow \text{SOS rules}
Stochastic Process Algebra

- Models are constructed from components which engage in activities.

\[(\alpha, r).P\]

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

- Models are constructed from components which engage in activities.

\[(\alpha, r).P\]

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

- Models are constructed from components which engage in activities.

\[(\alpha, r).P\]

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

\[
\begin{array}{c}
\text{SPA MODEL} \quad \text{SOS rules} \quad \text{LABELLED TRANSITION SYSTEM} \quad \text{state transition diagram} \quad \text{CTMC Q}
\end{array}
\]
PEPA

\[ S ::= (\alpha, r) S | S + S | A \]
\[ P ::= S | P \oplus L P | P/L \]
PEPA

\[ S ::= (\alpha, r).S \mid S + S \mid A \]

\[ P ::= S \mid P \Leftrightarrow P \mid P/L \]

**PREFIX:** \((\alpha, r).S\) designated first action
Lumpability

Strong Equivalence in PEPA

Contextual Lumpability

Conclusions

PEPA

\[ S ::= (\alpha, r).S | S + S | A \]
\[ P ::= S | P \otimes_P P | P/L \]

PREFIX: \((\alpha, r).S\) designated first action

CHOICE: \(S + S\) competing components
PEPA

\[ S ::= (\alpha, r).S \mid S + S \mid A \]
\[ P ::= S \mid P \lhd P \mid P/L \]

**PREFIX:** \((\alpha, r).S\) designated first action

**CHOICE:** \(S + S\) competing components

**CONSTANT:** \(A \overset{\text{def}}{=} S\) assigning names
PEPA

\[ S ::= (\alpha, r).S \mid S + S \mid A \]
\[ P ::= S \mid P \mathbin{\uplus} P \mid P/L \]

PREFIX: \((\alpha, r).S\) designated first action

CHOICE: \(S + S\) competing components

CONSTANT: \(A \overset{\text{def}}{=} S\) assigning names

COOPERATION: \(P \mathbin{\uplus}_L P\) \(\alpha \notin L\) individual actions
\(\alpha \in L\) shared actions
PEPA

\[ S ::= (\alpha, r).S | S + S | A \]
\[ P ::= S | P \Join L P | P/L \]

**PREFIX:** \((\alpha, r).S\) designated first action

**CHOICE:** \(S + S\) competing components

**CONSTANT:** \(A \overset{\text{def}}{=} S\) assigning names

**COOPERATION:** \(P \Join_L P\) \(\alpha \notin L\) individual actions
\(\alpha \in L\) shared actions

**HIDING:** \(P/L\) abstraction \(\alpha \in L \Rightarrow \alpha \rightarrow \tau\)
Process algebras often define equivalence relations in terms of *bisimulations*, based on the notion of *observability*.
Process algebras often define equivalence relations in terms of bisimulations, based on the notion of observability.
Process algebras often define equivalence relations in terms of bisimulations, based on the notion of observability.
Equivalence Relations

Process algebras often define equivalence relations in terms of **bisimulations**, based on the notion of **observability**.

In PEPA **observation** is assumed to include the ability to record **timing** information over a number of runs.
Process algebras often define equivalence relations in terms of **bisimulations**, based on the notion of **observability**.

In PEPA **observation** is assumed to include the ability to record **timing** information over a number of runs.
Equivalence Relations

Process algebras often define equivalence relations in terms of **bisimulations**, based on the notion of **observability**.

In PEPA **observation** is assumed to include the ability to record **timing** information over a number of runs.
Process algebras often define equivalence relations in terms of **bisimulations**, based on the notion of **observability**.

In PEPA **observation** is assumed to include the ability to record **timing** information over a number of runs.
Strong Equivalence in PEPA

**Strong Equivalence**

An equivalence relation \( \mathcal{R} \subseteq \mathcal{C} \times \mathcal{C} \) is a **strong equivalence** if whenever \((P, Q) \in \mathcal{R}\) then for all \(\alpha \in \mathcal{A}\) and for all \(S \in \mathcal{C}/\mathcal{R}\)

\[
q[P, S, \alpha] = q[Q, S, \alpha].
\]

where

\[
q[C_i, S, \alpha] = \sum_{C_j \in S} q(C_i, C_j, \alpha)
\]
Strong Equivalence

An equivalence relation $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$ is a strong equivalence if whenever $(P, Q) \in \mathcal{R}$ then for all $\alpha \in \mathcal{A}$ and for all $S \in \mathcal{C}/\mathcal{R}$

$$q[P, S, \alpha] = q[Q, S, \alpha].$$

where

$$q[C_i, S, \alpha] = \sum_{C_j \in S} q(C_i, C_j, \alpha)$$

Current action type preservation

A relation $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$ over PEPA components is current action type preserving if for all PEPA components $P, Q$ such that $(P, Q) \in \mathcal{R}$, $\mathcal{A}(P) = \mathcal{A}(Q)$. 

Contextuality

A relation $\mathcal{R} \subseteq C \times C$ over PEPA components is **contextual** if for all PEPA components $P, Q$ such that $(P, Q) \in \mathcal{R}$ and for all contexts $C[\cdot]$,

$$(C[P], C[Q]) \in \mathcal{R}$$

where a context is a term with a hole $[\cdot]$ defined by the grammar:

$$C[\cdot] ::= [\cdot] \mid [\cdot] \bowtie \\_ \_ P \mid P \bowtie \_ \_ [\cdot] \mid [\cdot]/L$$
Strong Equivalence and Lumpability

- Given this definition it is fairly straightforward to show that if we consider strong equivalence of states within a single model, it induces an ordinarily lumpable partition on the state space of the underlying Markov chain.
Strong Equivalence and Lumpability

- Given this definition it is fairly straightforward to show that if we consider **strong equivalence of states within a single model**, it induces an ordinarily lumpable partition on the state space of the underlying Markov chain.

- Moreover it can be shown that strong equivalence is **contextual**. (Indeed, it is a **congruence** with respect to all operators of PEPA including the dynamic ones.)
Strong Equivalence and Lumpability

- Given this definition it is fairly straightforward to show that if we consider strong equivalence of states within a single model, it induces an ordinarily lumpable partition on the state space of the underlying Markov chain.

- Moreover it can be shown that strong equivalence is contextual. (Indeed, it is a congruence with respect to all operators of PEPA including the dynamic ones.)

- This means that aggregation based on lumpability can be applied component by component.
Producer-Consumer Example

- We consider a system consisting of a **Producer** process and a **Consumer** process.
- The Producer **thinks**, **computes** and **transfers** results to the Consumer.
- Periodically **errors** occur during the computation and the process has to undergo **recovery** before it is ready to redo the computation.
- The **Consumer** enqueues the transferred jobs and **sends** them in batches.
- Jobs waiting in the queue can **spontaneously** generate further jobs.
- A maximum buffer size of $N$ jobs is set and, in case of saturation, further arrivals are lost.
Producer-Consumer Example: PEPA

Producer

\[ Q_{\text{Think}} \overset{\text{def}}{=} (\tau, \delta) \cdot Q_{\text{Compute}} \]
\[ Q_{\text{Compute}} \overset{\text{def}}{=} (\text{comp}, \varepsilon) \cdot Q_{\text{Send}} + (\tau, \varphi) \cdot Q_{\text{Error}} \]
\[ Q_{\text{Send}} \overset{\text{def}}{=} (tr, \eta) \cdot Q_{\text{Think}} \]
\[ Q_{\text{Error}} \overset{\text{def}}{=} (\tau, \xi) \cdot Q_{\text{Recovery}} \]
\[ Q_{\text{Recovery}} \overset{\text{def}}{=} (\tau, \delta) \cdot Q_{\text{Compute}} \]

Consumer

\[ P_{\text{Empty}} \overset{\text{def}}{=} (tr, \top) \cdot P_1 \]
\[ P_i \overset{\text{def}}{=} (\tau, i \mu) \cdot P_{i+1} + (tr, \top) \cdot P_{i+1} + (send, \gamma) \cdot P_{\text{Wait}} \quad 1 \leq i < N \]
\[ P_N \overset{\text{def}}{=} (tr, \top) \cdot P_N + (send, \gamma) \cdot P_{\text{Wait}} \]
\[ P_{\text{Wait}} \overset{\text{def}}{=} (\tau, \nu) \cdot P_{\text{Empty}} \]
Producer-Consumer Example: Producer
Producet-Consumer Example: Consumer

- $P_{Empty}$
- $P_1$
- $P_2$
- $P_N$
- $P_{Wait}$

Transitions:
- $\tau, \mu$
- $\tau, 2\mu$
- $tr, \top$
- $send, \gamma$
- $\tau, \nu$

Example
Example

Producer-Consumer Example: System

The whole system is modelled by:

\[ S \overset{\text{def}}{=} P_{Empty} \{ tr \} Q_{Think} \cdot \]
Example revisited: Original producer

We can see that $Q_{\text{Think}}$ and $Q_{\text{Recovery}}$ are strongly equivalent and consequently reduce the representation of the producer.
We can see that $Q_{\text{Think}}$ and $Q_{\text{Recovery}}$ are strongly equivalent and consequently reduce the representation of the producer.
We can see that $Q_{Think}$ and $Q_{Recovery}$ are strongly equivalent and consequently reduce the representation of the producer.
Example revisited: Reduced producer

We can see that $Q_{Think}$ and $Q_{Recovery}$ are strongly equivalent and consequently reduce the representation of the producer.

No such reduction is possible for the Consumer.
Contextual Lumpability

We know that any partition generated by strong equivalence at the process algebra level gives rise to a lumpable partition in the underlying Markov chain.
Contextual Lumpability

We know that any partition generated by strong equivalence at the process algebra level gives rise to a lumpable partition in the underlying Markov chain.

But we also know that this is not always the largest lumping that could be achieved if we worked directly at the level of the Markov chain.
Contextual Lumpability

We know that any partition generated by strong equivalence at the process algebra level gives rise to a lumpable partition in the underlying Markov chain.

But we also know that this is not always the largest lumping that could be achieved if we worked directly at the level of the Markov chain.

On the other hand, the contextually of the relation — the fact that it is preserved by the static combinators of the language — has considerable advantages when working on model reduction.
Contextual Lumpability

We know that any partition generated by strong equivalence at the process algebra level gives rise to a lumpable partition in the underlying Markov chain.

But we also know that this is not always the largest lumping that could be achieved if we worked directly at the level of the Markov chain.

On the other hand, the contextually of the relation — the fact that it is preserved by the static combinators of the language — has considerable advantages when working on model reduction.

Therefore we seek to define a relation which maintains contextually but achieves greater lumping.
Contextual Lumpability: formal definition

**Contextual lumpability**, denoted $\equiv_\text{l}$, is the largest contextual, current action type preserving, lumpable relation over PEPA terms.
Contextual Lumpability: formal definition

Contextual lumpability, denoted $\simeq_l$, is the largest contextual, current action type preserving, lumpable relation over PEPA terms.
Contextual Lumpability: formal definition

Contextual lumpability, denoted \( \equiv_l \), is the largest contextual, current action type preserving, lumpable relation over PEPA terms.
Contextual Lumpability: formal definition

Contextual lumpability

Contextual lumpability, denoted $\cong_I$, is the largest contextual, current action type preserving, lumpable relation over PEPA terms.
Lumpable Bisimulation

We can consider contextual lumpability as a target or template for what we would like to achieve, but it is not the definition of equivalence in the usual process algebra style of bisimulation based on observability.
Lumpable Bisimulation

We can consider contextual lumpability as a target or template for what we would like to achieve, but it is not the definition of equivalence in the usual process algebra style of bisimulation based on observability.

Thus we seek to define a bisimulation that can be seen to fit the template of contextual lumpability.
Lumpable Bisimulation: formal definition

An equivalence relation over PEPA components, $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$, is a lumpable bisimulation if whenever $(P, Q) \in \mathcal{R}$ then for all $\alpha \in \mathcal{A}$ and for all $S \in \mathcal{C}/\mathcal{R}$ such that

- either $\alpha \neq \tau$,
- or $\alpha = \tau$ and $P, Q \notin S$,

it holds

$$q[P, S, \alpha] = q[Q, S, \alpha].$$
Remarks

Note that this definition allows arbitrary $\tau$ activities between components belonging to the same equivalence class.
Remarks

- Note that this definition allows arbitrary $\tau$ activities between components belonging to the same equivalence class.
- Thus it can be regarded as coarser than strong equivalence.
Remarks

- Note that this definition allows arbitrary $\tau$ activities between components belonging to the same equivalence class.

- Thus it can be regarded as coarser than strong equivalence.

- However, it has weaker congruence properties because although it is contextual, unlike strong equivalence it is not preserved by choice.
Remarks

- Note that this definition allows arbitrary $\tau$ activities between components belonging to the same equivalence class.

- Thus it can be regarded as coarser than strong equivalence.

- However, it has weaker congruence properties because although it is contextual, unlike strong equivalence it is not preserved by choice.

- Nevertheless for practical purposes this is sufficient for our needs.
Lumpable Bisimilarity

Proposition
Let $I$ be a set of indices and $\mathcal{R}_i$ be a lumpable bisimulation for all $i \in I$. Then the transitive closure of their union, $\mathcal{R} = (\bigcup_{i \in I} \mathcal{R}_i)^*$, is also a lumpable bisimulation.
Lumpable Bisimilarity

**Proposition**

Let $I$ be a set of indices and $\mathcal{R}_i$ be a lumpable bisimulation for all $i \in I$. Then the transitive closure of their union, $\mathcal{R} = (\bigcup_{i \in I} \mathcal{R}_i)^*$, is also a lumpable bisimulation.

**Lumpable bisimilarity**

Two PEPA components $P$ and $Q$ are lumpably bisimilar, written $P \approx_I Q$, if $(P, Q) \in \mathcal{R}$ for some lumpable bisimulation $\mathcal{R}$, i.e.,

$$\approx_I = \bigcup \{ \mathcal{R} \mid \mathcal{R} \text{ is a lumpable bisimulation} \}.$$  

$\approx_I$ is called lumpable bisimilarity and it is the largest symmetric lumpable bisimulation over PEPA components.
It remains to show that lumpable bisimilarity is a characterisation of contextual lumpability.
Lumpable Bisimilarity and Contextual Lumpability

It remains to show that lumpable bisimilarity is a characterisation of contextual lumpability.

**Theorem**

Let $P$ and $Q$ be two PEPA components.  
It holds that: $P \approx_1 Q$ if and only if $P \approx_1 Q$. 
Example revisited: Original producer
Example revisited: Original producer

We can see that $Q_{\text{Think}} \approx_1 Q_{\text{Recovery}}$ and hence they belong to the same equivalence class, say $[Q']$. 
Example revisited: Reduced producer

We can see that $Q_{Think} \approx l Q_{Recovery}$ and hence they belong to the same equivalence class, say $[Q']$. 
Example revisited: Original consumer

Here we observe that $P_1 \approx_l P_2 \approx_l \cdots \approx_l P_N$ and let $[P']$ be the associated equivalence class.
Example revisited: Original consumer

Here we observe that $P_1 \approx_1 P_2 \approx_1 \cdots \approx_1 P_N$ and let $[P']$ be the associated equivalence class.
Example revisited: Reduced consumer

Here we observe that $P_1 \approx_I P_2 \approx_I \cdots \approx_I P_N$ and let $[P']$ be the associated equivalence class.
Algorithm

We take advantage of an existing algorithm \textsc{Lumpability}(G = (D, Δ, W), I), developed by Valmari and Franceschinis in 2010 (TACAS 2010).
Algorithm

- We take advantage of an existing algorithm $\text{Lumpability}(G = (D, \Delta, W), I)$, developed by Valmari and Franceschinis in 2010 (TACAS 2010).

- $\text{Lumpability}(G = (D, \Delta, W), I)$ computes the largest equivalence relation $R \subseteq I$ over the derivative set $D$ compatible with the derivation graph $G$. 

**Algorithm**

- We take advantage of an existing algorithm $\text{Lumpability}(G = (D, \Delta, W), I)$, developed by Valmari and Franceschinis in 2010 (TACAS 2010).

- $\text{Lumpability}(G = (D, \Delta, W), I)$ computes the largest equivalence relation $R \subseteq I$ over the derivative set $D$ compatible with the derivation graph $G$.

- $\text{Contextual\_Lumpability}(\{G_\alpha^{D}\}_{\alpha \in A \setminus \{\tau\}}, \hat{G}_\tau^{D})$ returns the relation $\approx_I \cap (D \times D)$, i.e., the largest lumpable bisimulation over $D$. 
Algorithm

\textsc{Contextual\_Lumpability}(\{G^D_\alpha\}_{\alpha \in A\setminus\{\tau\}}, \hat{G}^D_\tau)

\textbf{Require:} \ D \ finite \ and \ \ D = ds(\mathcal{D});
\begin{align*}
  & R = \emptyset; \\
  & \mathcal{R} = \mathcal{D} \times \mathcal{D}; \\
  \text{while} \ R \neq \mathcal{R} \ \text{do} \\
  & \quad R = \mathcal{R}; \\
  & \quad \text{for all} \ \alpha \in A \setminus \{\tau\} \ \text{do} \\
  & \quad \quad \mathcal{R} = \text{Lumpability}(G^D_\alpha, \mathcal{R}); \\
  & \quad \text{end for} \\
  & \quad \mathcal{R} = \text{Lumpability}(\hat{G}^D_\tau, \mathcal{R}); \\
  \text{end while} \\
  \text{return} \ \mathcal{R};
\end{align*}
Conclusions

- We have introduced a relaxed form of lumpability for PEPA models, termed **contextual lumpability**.
- We also defined a new equivalence relation for PEPA models, termed **lumpable bisimilarity**.
- We showed that lumpable bisimilarity is a **characterisation** of contextual lumpability and induces the largest contextual lumping of the underlying Markov chain of a PEPA model.
- Finally we provided an **algorithm** for lumpable bisimilarity.