Calculi for Biological Systems

Jane Hillston and Federica Ciocchetta.
LFCS,
University of Edinburgh

5th June 2008

Outline

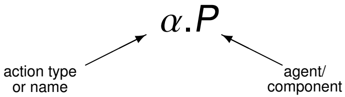
- Process Calculi
  - SPA
- SPA for Systems Biology
- Bio-PEPA
  - Model definition
  - The syntax and semantics
  - Equivalences and Analysis
- Examples
  - Genetic network with negative feedback loop
  - Goldbeter's model
- Conclusions

Outline

- Process Calculi
  - SPA
- SPA for Systems Biology
- Bio-PEPA
  - Model definition
  - The syntax and semantics
  - Equivalences and Analysis
- Examples
  - Genetic network with negative feedback loop
  - Goldbeter's model
- Conclusions

Process Algebra

- Models consist of agents which engage in actions.



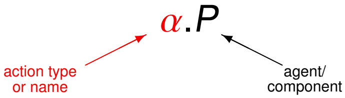
$\alpha.P$

action type or name

agent/ component

Process Algebra

- Models consist of agents which engage in actions.



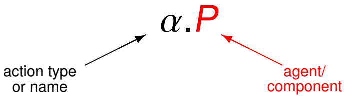
$\alpha.P$

action type
or name

agent/
component

Process Algebra

- Models consist of agents which engage in actions.



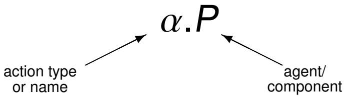
$$\alpha.P$$

action type or name

agent/ component

Process Algebra

- Models consist of agents which engage in actions.

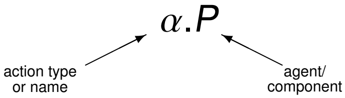


$$\alpha.P$$

- 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.



$\alpha.P$

action type or name

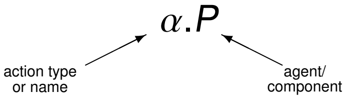
agent/ component

- 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.



$\alpha.P$

action type or name

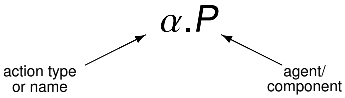
agent/ component

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

Process algebra model $\xrightarrow{\text{SOS rules}}$

Process Algebra

- Models consist of agents which engage in actions.



$\alpha.P$

action type or name

agent/ component

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

$$\text{Process algebra model} \xrightarrow{\text{SOS rules}} \text{Labelled transition system}$$

Example

Consider a web server which offers html pages for download:

$$Server \overset{def}{=} get.download.rel.Server$$

Example

Consider a web server which offers html pages for download:

$$Server \overset{def}{=} get.download.rel.Server$$

Clients are web browsers, in a domain with a local cache of frequently requested pages. Thus any display request might result in an access to the server or in a page being loaded from the cache.

$$Browser \overset{def}{=} display.(cache.Browser + get.download.rel.Browser)$$

Example

Consider a web server which offers html pages for download:

$$Server \overset{def}{=} get.download.rel.Server$$

Clients are web browsers, in a domain with a local cache of frequently requested pages. Thus any display request might result in an access to the server or in a page being loaded from the cache.

$$Browser \overset{def}{=} display.(cache.Browser + get.download.rel.Browser)$$

A simple version of the Web can be considered to be the interaction of these components:

$$WEB \overset{def}{=} (Browser \parallel Browser) \mid Server$$

Dynamic behaviour

- The behaviour of a model is dictated by the semantic rules governing the combinators of the language.

Dynamic behaviour

- The behaviour of a model is dictated by the semantic rules governing the combinators of the language.
- The possible evolutions of a model are captured by applying these rules exhaustively, generating a labelled transition system.

Dynamic behaviour

- The behaviour of a model is dictated by the semantic rules governing the combinators of the language.
- The possible evolutions of a model are captured by applying these rules exhaustively, generating a labelled transition system.
- This can be viewed as a graph in which each node is a state of the model (comprised of the local states of each of the components) and the arcs represent the actions which can cause the move from one state to another.

Dynamic behaviour

- The behaviour of a model is dictated by the semantic rules governing the combinators of the language.
- The possible evolutions of a model are captured by applying these rules exhaustively, generating a labelled transition system.
- This can be viewed as a graph in which each node is a state of the model (comprised of the local states of each of the components) and the arcs represent the actions which can cause the move from one state to another.
- The language is also equipped with observational equivalence which can be used to compare models.

Dynamic behaviour

$$Browser \overset{def}{=} display.(cache.Browser \;+\; get.download.rel.Browser)$$

Dynamic behaviour

$$Browser \overset{def}{=} display.(cache.Browser \;+\; get.download.rel.Browser)$$

$$\frac{}{(\alpha, r).P \xrightarrow{(\alpha,r)} P}$$

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P + Q \xrightarrow{(\alpha,r)} P'}$$

$$\frac{Q \xrightarrow{(\alpha,r)} Q'}{P + Q \xrightarrow{(\alpha,r)} Q'}$$

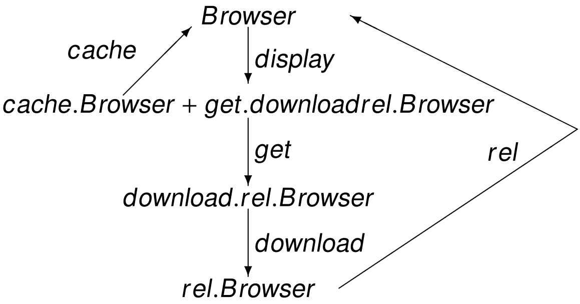
Dynamic behaviour

$$Browser \overset{def}{=} display.(cache.Browser \quad + \quad get.download.rel.Browser)$$

$$\frac{}{(\alpha, r).P \xrightarrow{(\alpha,r)} P}$$

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P + Q \xrightarrow{(\alpha,r)} P'}$$

$$\frac{Q \xrightarrow{(\alpha,r)} Q'}{P + Q \xrightarrow{(\alpha,r)} Q'}$$



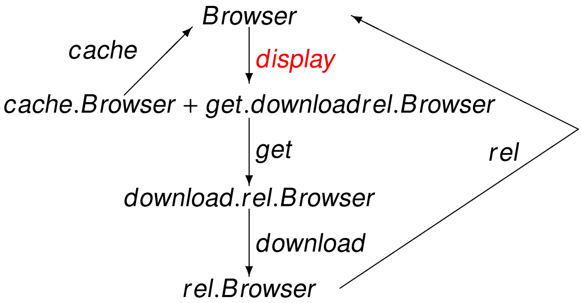
Dynamic behaviour

$$Browser \overset{def}{=} display.(cache.Browser \; + \; get.download.rel.Browser)$$

$$\frac{}{(\alpha,r).P \xrightarrow{(\alpha,r)} P}$$

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P+Q \xrightarrow{(\alpha,r)} P'}$$

$$\frac{Q \xrightarrow{(\alpha,r)} Q'}{P+Q \xrightarrow{(\alpha,r)} Q'}$$



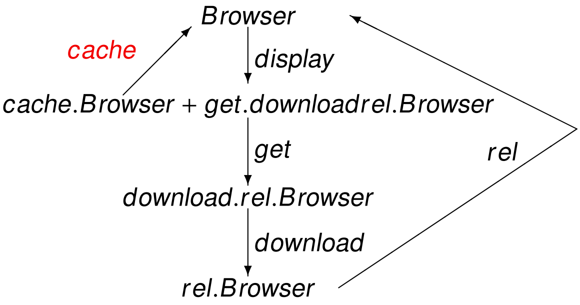
Dynamic behaviour

$$Browser \overset{def}{=} display.(cache.Browser \;+\; get.download.rel.Browser)$$

$$\frac{}{(\alpha, r).P \xrightarrow{(\alpha,r)} P}$$

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P + Q \xrightarrow{(\alpha,r)} P'}$$

$$\frac{Q \xrightarrow{(\alpha,r)} Q'}{P + Q \xrightarrow{(\alpha,r)} Q'}$$



Browser

cache

display

cache.Browser + get.downloadrel.Browser

get

rel

download.rel.Browser

download

rel.Browser

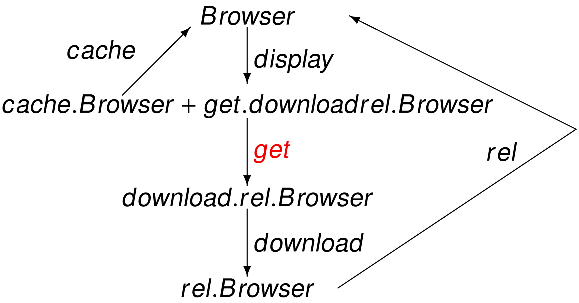
Dynamic behaviour

$$Browser \overset{def}{=} display.(cache.Browser \; + \; get.download.rel.Browser)$$

$$\frac{}{(\alpha, r).P \xrightarrow{(\alpha,r)} P}$$

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P + Q \xrightarrow{(\alpha,r)} P'}$$

$$\frac{Q \xrightarrow{(\alpha,r)} Q'}{P + Q \xrightarrow{(\alpha,r)} Q'}$$



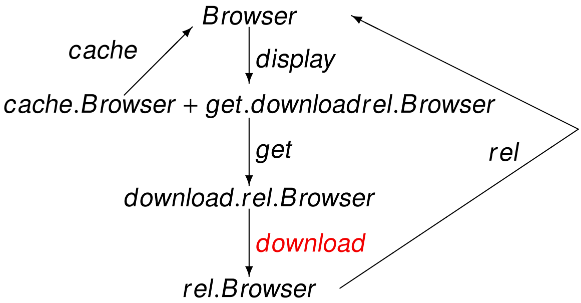
Dynamic behaviour

$$Browser \overset{def}{=} display.(cache.Browser \;+\; get.\color{red}{download.}\color{black}{rel.Browser})$$

$$\color{red}{\overline{(\alpha, r).P \xrightarrow{(\alpha,r)} P}}$$

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P + Q \xrightarrow{(\alpha,r)} P'}$$

$$\frac{Q \xrightarrow{(\alpha,r)} Q'}{P + Q \xrightarrow{(\alpha,r)} Q'}$$



Browser

cache

display

cache.Browser + get.downloadrel.Browser

get

rel

download.rel.Browser

download

rel.Browser

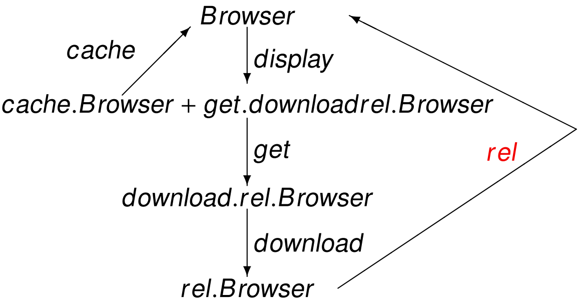
Dynamic behaviour

$$Browser \overset{def}{=} display.(cache.Browser \;+\; get.download.rel.Browser)$$

$$\frac{}{(\alpha, r).P \xrightarrow{(\alpha,r)} P}$$

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P + Q \xrightarrow{(\alpha,r)} P'}$$

$$\frac{Q \xrightarrow{(\alpha,r)} Q'}{P + Q \xrightarrow{(\alpha,r)} Q'}$$



Browser

cache

display

cache.Browser + get.downloadrel.Browser

get

rel

download.rel.Browser

download

rel.Browser

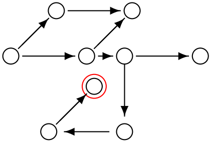
Qualitative Analysis

- The labelled transition system underlying a process algebra model can be used for functional verification e.g.: reachability analysis, specification matching and model checking.

Qualitative Analysis

- The labelled transition system underlying a process algebra model can be used for functional verification e.g.: reachability analysis, specification matching and model checking.

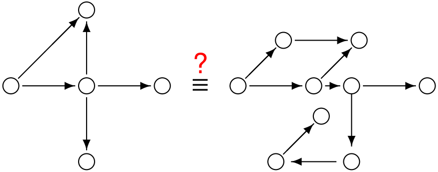
Will the system arrive in a particular state?



Qualitative Analysis

- The labelled transition system underlying a process algebra model can be used for functional verification e.g.: reachability analysis, specification matching and model checking.

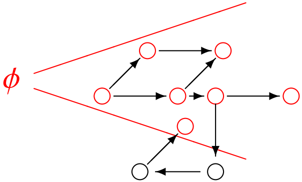
Does system behaviour match its specification?



Qualitative Analysis

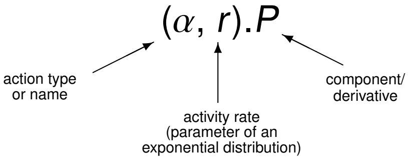
- The labelled transition system underlying a process algebra model can be used for functional verification e.g.: reachability analysis, specification matching and model checking.

Does a given property ϕ hold within the system?



Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$$(\alpha, r).P$$

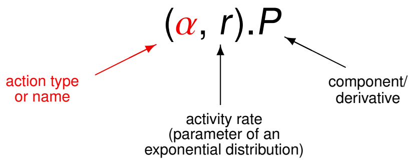
action type or name

activity rate (parameter of an exponential distribution)

component/ derivative

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$$(\alpha, r).P$$

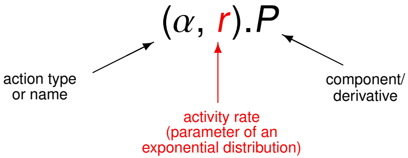
action type or name

activity rate (parameter of an exponential distribution)

component/ derivative

Stochastic Process Algebra

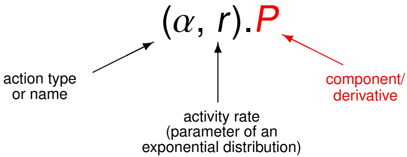
- Models are constructed from components which engage in activities.



$(\alpha, r).P$

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$$(\alpha, r).P$$

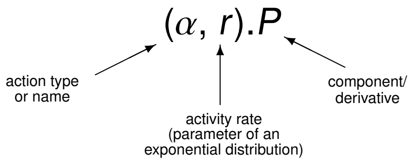
action type
or name

activity rate
(parameter of an
exponential distribution)

component/
derivative

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

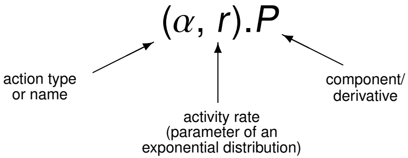


$(\alpha, r).P$

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

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

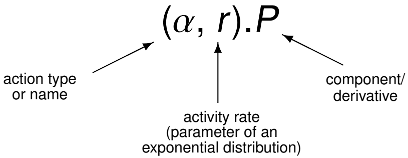


- The language is used to generate a CTMC for performance modelling.

SPA
MODEL

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$(\alpha, r).P$

action type or name

activity rate (parameter of an exponential distribution)

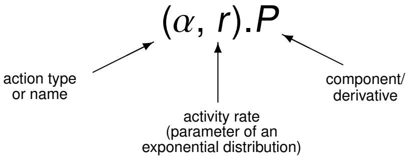
component/ derivative

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

SPA MODEL $\xrightarrow{\text{SOS rules}}$

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



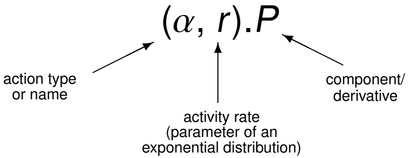
$$(\alpha, r).P$$

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

$$\text{SPA MODEL} \xrightarrow{\text{SOS rules}} \text{LABELLED TRANSITION SYSTEM}$$

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



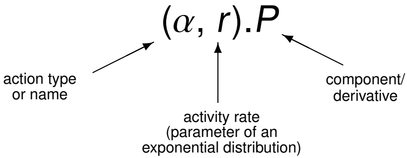
$$(\alpha, r).P$$

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

SPA MODEL $\xrightarrow{\text{SOS rules}}$ LABELLED TRANSITION SYSTEM $\xrightarrow[\text{diagram}]{\text{state transition}}$

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



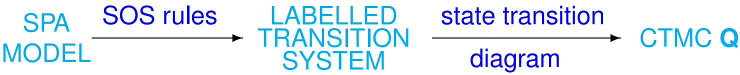
$(\alpha, r).P$

action type or name

activity rate (parameter of an exponential distribution)

component/ derivative

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



SPA MODEL $\xrightarrow[\text{}]{\text{SOS rules}}$ LABELLED TRANSITION SYSTEM $\xrightarrow[\text{diagram}]{\text{state transition}}$ CTMC **Q**

PEPA

$$
\begin{array}{lcl}
S & ::= & (\alpha, r).S \mid S + S \mid A \\
P & ::= & S \mid P \underset{L}{\bowtie} P \mid P/L
\end{array}
$$

PEPA

$$
\begin{array}{rcl}
S & ::= & (\alpha, r).S \mid S + S \mid A \\
P & ::= & S \mid P \underset{L}{\bowtie} P \mid P/L
\end{array}
$$

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

PEPA

$$\begin{array}{rcl} S & ::= & (\alpha, r).S \mid S + S \mid A \\ P & ::= & S \mid P \underset{L}{\bowtie} P \mid P/L \end{array}$$

PREFIX:	$(\alpha, r).S$	designated first action
CHOICE:	$S + S$	competing components (race policy)

PEPA

$$
\begin{array}{rcl}
S & ::= & (\alpha, r).S \mid S + S \mid A \\
P & ::= & S \mid P \underset{L}{\bowtie} P \mid P/L
\end{array}
$$

PREFIX:	$(\alpha, r).S$	designated first action
CHOICE:	$S + S$	competing components (race policy)
CONSTANT:	$A \overset{def}{=} S$	assigning names

PEPA

$$\begin{array}{rcl} S & ::= & (\alpha, r).S \mid S + S \mid A \\ P & ::= & S \mid P \underset{L}{\bowtie} P \mid P/L \end{array}$$

PREFIX:	$(\alpha, r).S$	designated first action
CHOICE:	$S + S$	competing components (race policy)
CONSTANT:	$A \overset{def}{=} S$	assigning names
COOPERATION:	$P \underset{L}{\bowtie} P$	$\alpha \notin L$ concurrent activity (*individual actions*) $\alpha \in L$ cooperative activity (*shared actions*)

PEPA

$$\begin{array}{rcl} S & ::= & (\alpha, r).S \mid S + S \mid A \\ P & ::= & S \mid P \underset{L}{\bowtie} P \mid P/L \end{array}$$

PREFIX:	$(\alpha, r).S$	designated first action
CHOICE:	$S + S$	competing components (race policy)
CONSTANT:	$A \overset{def}{=} S$	assigning names
COOPERATION:	$P \underset{L}{\bowtie} P$	$\alpha \notin L$ concurrent activity (*individual actions*) $\alpha \in L$ cooperative activity (*shared actions*)

PEPA

$$
\begin{aligned}
S &::= (\alpha, r).S \mid S + S \mid A \\
P &::= S \mid P \underset{L}{\bowtie} P \mid P/L
\end{aligned}
$$

PREFIX:	$(\alpha, r).S$	designated first action
CHOICE:	$S + S$	competing components (race policy)
CONSTANT:	$A \overset{def}{=} S$	assigning names
COOPERATION:	$P \underset{L}{\bowtie} P$	$\alpha \notin L$ concurrent activity (*individual actions*) $\alpha \in L$ cooperative activity (*shared actions*)
HIDING:	P/L	abstraction $\alpha \in L \Rightarrow \alpha \rightarrow \tau$

Example revisited

The behaviour of the server is the same but now quantitative information is recorded for each operation:

$$Server \overset{def}{=} (get, \top).(download, \mu).(rel, \top).Server$$

Example revisited

The behaviour of the server is the same but now quantitative information is recorded for each operation:

$$Server \overset{def}{=} (get, \top).(download, \mu).(rel, \top).Server$$

In addition to duration we also incorporate information about the relative frequencies of the different actions which take place after a display request:

$$\begin{aligned} Browser \overset{def}{=} & (display, p_1\lambda).(cache, m).Browser \ + \\ & (display, p_2\lambda).(get, g).(download, \top).(rel, r).Browser \end{aligned}$$

Example revisited

The behaviour of the server is the same but now quantitative information is recorded for each operation:

$$Server \overset{def}{=} (get, \top).(download, \mu).(rel, \top).Server$$

In addition to duration we also incorporate information about the relative frequencies of the different actions which take place after a display request:

$$\begin{aligned} Browser \overset{def}{=} & \ (display, p_1\lambda).(cache, m).Browser \ + \\ & \ (display, p_2\lambda).(get, g).(download, \top).(rel, r).Browser \end{aligned}$$

The configuration is recorded as before; using the PEPA cooperation the actions which must be shared are explicitly named:

$$WEB \overset{def}{=} (Browser \parallel Browser) \underset{L}{\bowtie} Server \qquad L = \{get, download, rel\}$$

Example revisited

The behaviour of the server is the same but now quantitative information is recorded for each operation:

$$Server \overset{def}{=} (get, \top).(download, \mu).(rel, \top).Server$$

In addition to duration we also incorporate information about the relative frequencies of the different actions which take place after a display request:

$$\begin{aligned} Browser \overset{def}{=} & \; (display, p_1\lambda).(cache, m).Browser \;+ \\ & \; (display, p_2\lambda).(get, g).(download, \top).(rel, r).Browser \end{aligned}$$

The configuration is recorded as before; using the PEPA cooperation the actions which must be shared are explicitly named:

$$WEB \overset{def}{=} (Browser \parallel Browser) \underset{L}{\bowtie} Server \qquad L = \{get, download, rel\}$$

Integrated analysis

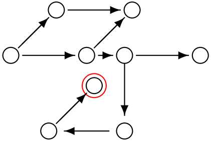
- Qualitative verification can now be complemented by quantitative verification:

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?

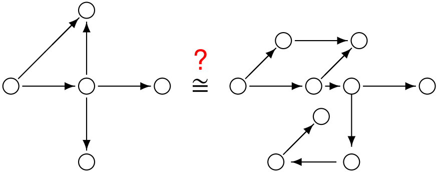


Integrated analysis

- Qualitative verification can now be complemented by quantitative verification:

Specification matching

With what probability does system behaviour match its specification?

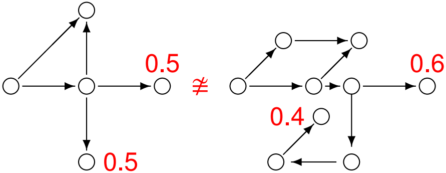


Integrated analysis

- Qualitative verification can now be complemented by quantitative verification:

Specification matching

Does the "*frequency profile*" of the system match that of the specification?

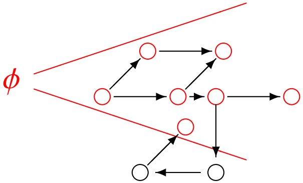


Integrated analysis

- Qualitative verification can now be complemented by quantitative verification:

Model checking

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

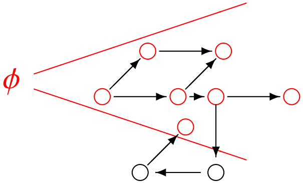


Integrated analysis

- Qualitative verification can now be complemented by quantitative verification:

Model checking

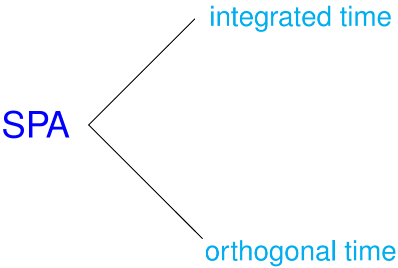
For a given starting state
how long is it until
a given property ϕ holds?



SPA Languages

SPA

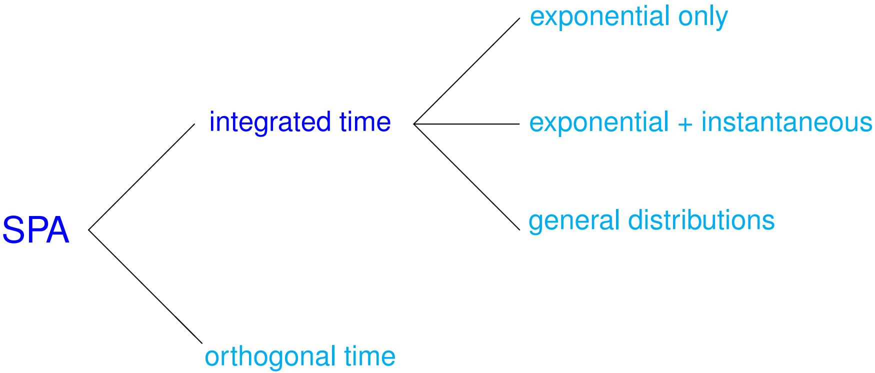
SPA Languages


SPA

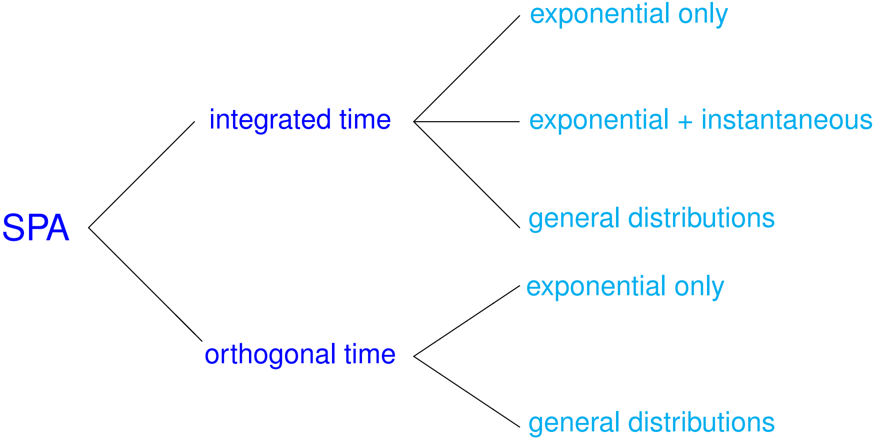
integrated time

orthogonal time

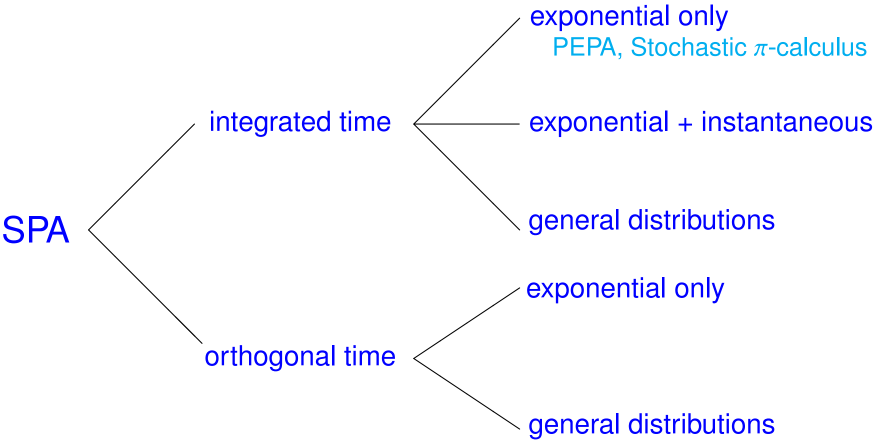
SPA Languages



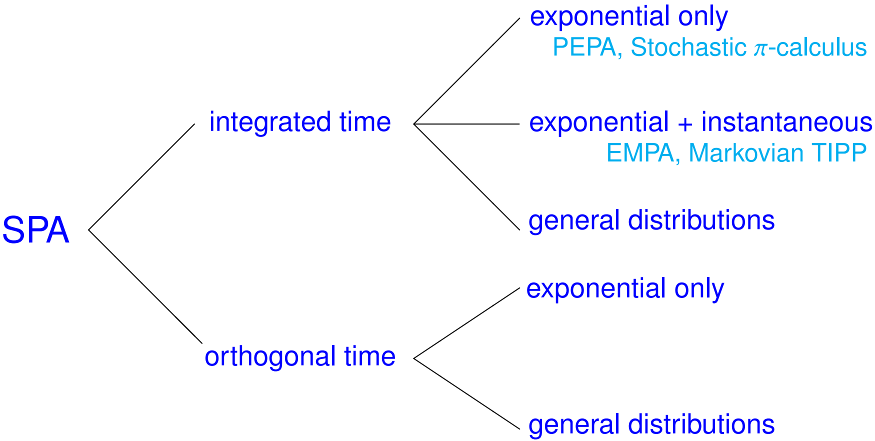
SPA Languages



SPA Languages



SPA Languages



SPA

integrated time

exponential only

PEPA, Stochastic π-calculus

exponential + instantaneous

EMPA, Markovian TIPP

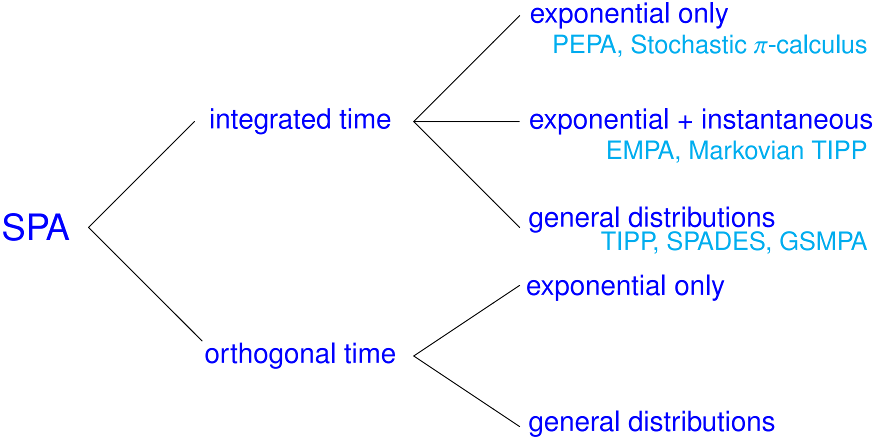
general distributions

orthogonal time

exponential only

general distributions

SPA Languages



SPA

integrated time

exponential only

PEPA, Stochastic π-calculus

exponential + instantaneous

EMPA, Markovian TIPP

general distributions

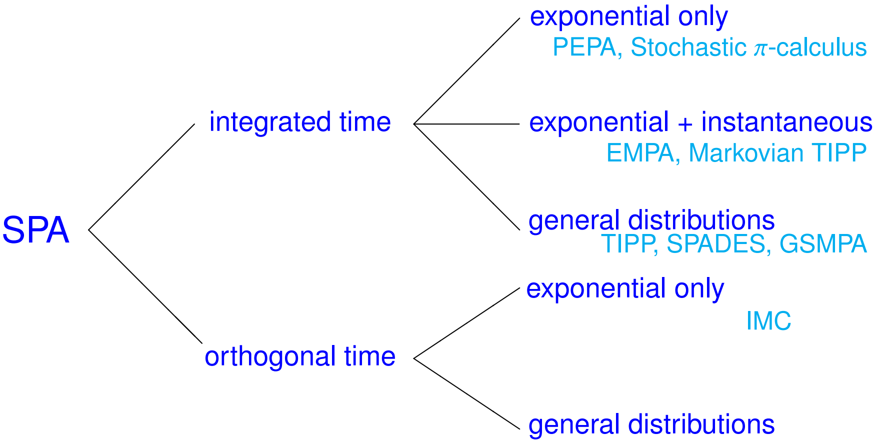
TIPP, SPADES, GSMPA

orthogonal time

exponential only

general distributions

SPA Languages



SPA

integrated time

exponential only

PEPA, Stochastic π-calculus

exponential + instantaneous

EMPA, Markovian TIPP

general distributions

TIPP, SPADES, GSMPA

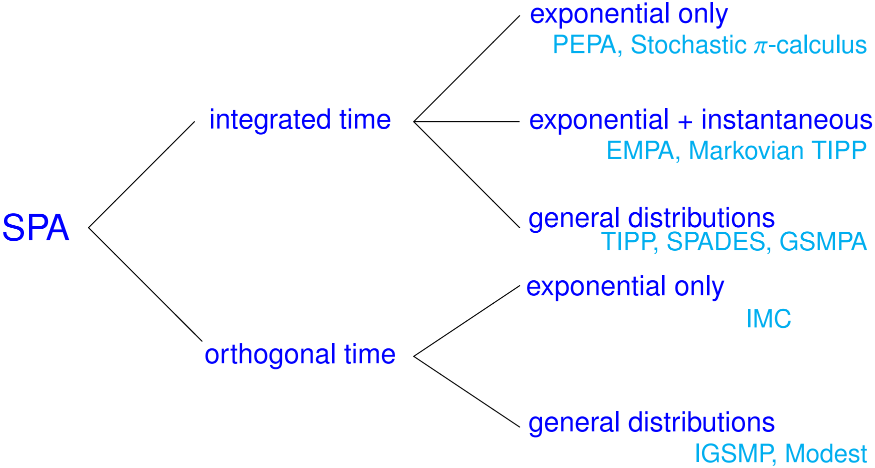
orthogonal time

exponential only

IMC

general distributions

SPA Languages



SPA

integrated time

exponential only

PEPA, Stochastic π-calculus

exponential + instantaneous

EMPA, Markovian TIPP

general distributions

TIPP, SPADES, GSMPA

orthogonal time

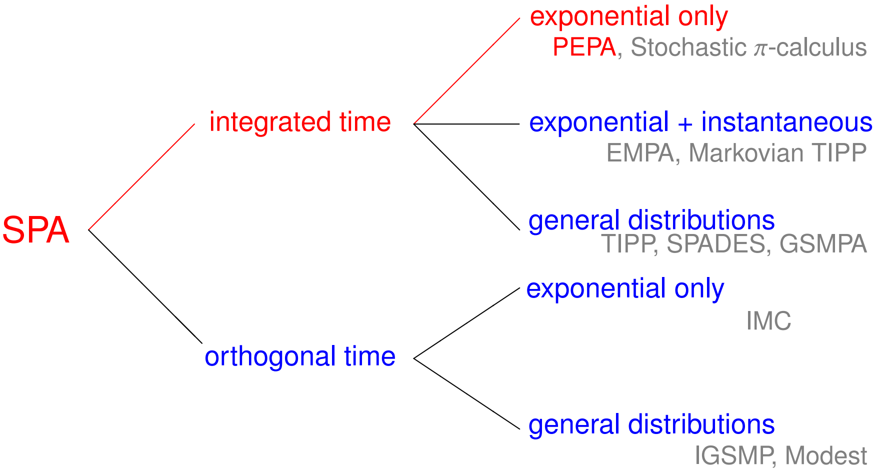
exponential only

IMC

general distributions

IGSMP, Modest

SPA Languages



SPA

integrated time

exponential only

PEPA, Stochastic π-calculus

exponential + instantaneous

EMPA, Markovian TIPP

general distributions

TIPP, SPADES, GSMPA

orthogonal time

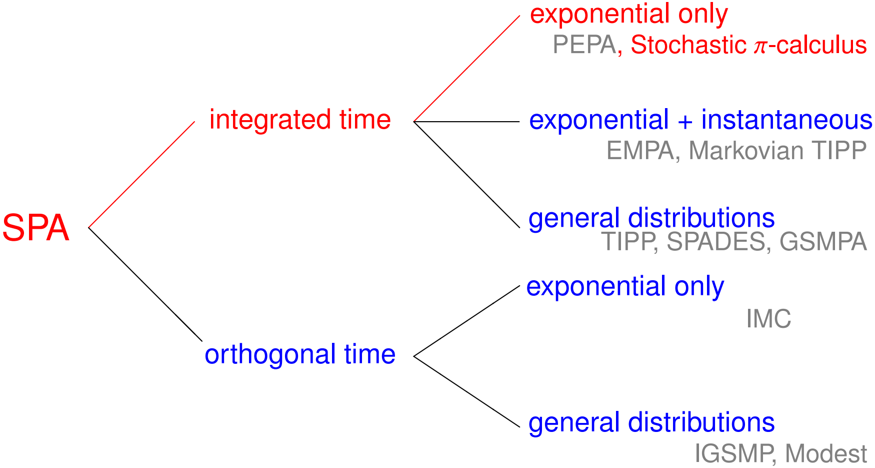
exponential only

IMC

general distributions

IGSMP, Modest

SPA Languages



SPA

integrated time

exponential only
PEPA, Stochastic π-calculus

exponential + instantaneous
EMPA, Markovian TIPP

general distributions
TIPP, SPADES, GSMPA

orthogonal time

exponential only
IMC

general distributions
IGSMP, Modest

The Importance of Being Exponential

$$\begin{tikzcd}
 & (\alpha, r).\mathit{Stop} \parallel (\beta, s).\mathit{Stop} \arrow[dl, "{(\alpha, r)}"'] \arrow[dr, "{(\beta, s)}"] & \\
\mathit{Stop} \parallel (\beta, s).\mathit{Stop} \arrow[dr, "{(\beta, s)}"'] & & (\alpha, r).\mathit{Stop} \parallel \mathit{Stop} \arrow[dl, "{(\alpha, r)}"] \\
 & \mathit{Stop} \parallel \mathit{Stop} &
\end{tikzcd}$$

The Importance of Being Exponential

$$\begin{tikzcd}
& (\alpha, r).\mathit{Stop} \parallel (\beta, s).\mathit{Stop} \arrow[dl, "{(\alpha, r)}"'] \arrow[dr, "{(\beta, s)}"] & \\
\mathit{Stop} \parallel (\beta, s).\mathit{Stop} \arrow[dr, "{(\beta, s)}"'] & & (\alpha, r).\mathit{Stop} \parallel \mathit{Stop} \arrow[dl, "{(\alpha, r)}"] \\
& \mathit{Stop} \parallel \mathit{Stop} &
\end{tikzcd}$$

The Importance of Being Exponential

$$\begin{tikzcd}
 & (\alpha, r).Stop \parallel (\beta, s).Stop \arrow[dl, "{\color{red}(\alpha, r)}"'] \arrow[dr, "{\color{red}(\beta, s)}"] & \\
Stop \parallel (\beta, s).Stop \arrow[dr, "(\beta, s)"'] & & (\alpha, r).Stop \parallel Stop \arrow[dl, "(\alpha, r)"] \\
 & Stop \parallel Stop &
\end{tikzcd}$$

The Importance of Being Exponential

$$\begin{tikzcd}
 & (\alpha, r).\mathit{Stop} \parallel (\beta, s).\mathit{Stop} \arrow[dl, "{(\alpha, r)}"'] \arrow[dr, "{(\beta, s)}"] \arrow[dd, red, "{(\alpha, r)}"', "{(\beta, s)}"] & \\
\mathit{Stop} \parallel (\beta, s).\mathit{Stop} \arrow[dr, "{(\beta, s)}"'] & & (\alpha, r).\mathit{Stop} \parallel \mathit{Stop} \arrow[dl, "{(\alpha, r)}"] \\
 & \mathit{Stop} \parallel \mathit{Stop} &
\end{tikzcd}$$

The Importance of Being Exponential

$$\begin{tikzcd}
 & (\alpha, r).\mathit{Stop} \parallel (\beta, s).\mathit{Stop} \arrow[dl, "(\alpha, r)"'] \arrow[dr, "(\beta, s)"] & \\
\mathit{Stop} \parallel (\beta, s).\mathit{Stop} \arrow[dr, "(\beta, s)"'] & & (\alpha, r).\mathit{Stop} \parallel \mathit{Stop} \arrow[dl, "(\alpha, r)"] \\
 & \mathit{Stop} \parallel \mathit{Stop} &
\end{tikzcd}$$

The Importance of Being Exponential

$$\begin{tikzcd}
 & (\alpha, r).Stop \parallel (\beta, s).Stop \arrow[dl, "{(\alpha, r)}"'] \arrow[dr, "{(\beta, s)}"] & \\
Stop \parallel (\beta, s).Stop \arrow[dr, "{(\beta, s)}"'] & & (\alpha, r).Stop \parallel Stop \arrow[dl, "{(\alpha, r)}"] \\
 & Stop \parallel Stop &
\end{tikzcd}$$

The Importance of Being Exponential

$$\begin{tikzcd}
 & (\alpha, r).Stop \parallel (\beta, s).Stop \arrow[dl, "(\alpha, r)"'] \arrow[dr, "(\beta, s)"] & \\
Stop \parallel (\beta, s).Stop \arrow[dr, "(\beta, s)"'] & & (\alpha, r).Stop \parallel Stop \arrow[dl, "(\alpha, r)"] \\
 & Stop \parallel Stop &
\end{tikzcd}$$

The Importance of Being Exponential

$$\begin{tikzcd}
 & (\alpha, r).\mathit{Stop} \parallel (\beta, s).\mathit{Stop} \arrow[dl, "{(\alpha, r)}"'] \arrow[dr, "{(\beta, s)}"] & \\
\mathit{Stop} \parallel (\beta, s).\mathit{Stop} \arrow[dr, "{(\beta, s)}"'] & & (\alpha, r).\mathit{Stop} \parallel \mathit{Stop} \arrow[dl, "{(\alpha, r)}"] \\
 & \mathit{Stop} \parallel \mathit{Stop} &
\end{tikzcd}$$

The memoryless property of the negative exponential distribution means that residual times do not need to be recorded.

The Importance of Being Exponential

$$\begin{tikzcd}
 & (\alpha, r).\mathit{Stop} \parallel (\beta, s).\mathit{Stop} \arrow[dl, "(\alpha, r)"'] \arrow[dr, "(\beta, s)"] & \\
\mathit{Stop} \parallel (\beta, s).\mathit{Stop} \arrow[dr, "(\beta, s)"'] & & (\alpha, r).\mathit{Stop} \parallel \mathit{Stop} \arrow[dl, "(\alpha, r)"] \\
 & \mathit{Stop} \parallel \mathit{Stop} &
\end{tikzcd}$$

We retain the expansion law of classical process algebra:

$$(\alpha, r).\mathit{Stop} \parallel (\beta, s).\mathit{Stop} = (\alpha, r).(\beta, s).(\mathit{Stop} \parallel \mathit{Stop}) + (\beta, s).(\alpha, r).(\mathit{Stop} \parallel \mathit{Stop})$$

only if the negative exponential distribution is assumed.

Parallel Composition

- Parellel composition is the basis of the compositionality in a process algebra

Parallel Composition

- Parellel composition is the basis of the compositionality in a process algebra — it defines which components interact and how.

Parallel Composition

- Parellel composition is the basis of the compositionality in a process algebra — it defines which components interact and how.
- In classical process algebra is it often associated with communication.

Parallel Composition

- Parellel composition is the basis of the compositionality in a process algebra — it defines which components interact and how.
- In classical process algebra is it often associated with communication.
- When the activities of the process algebra have a duration the definition of parallel composition becomes more complex.

Who Synchronises...?

Even within classical process algebras there is variation in the interpretation of parallel composition:

Who Synchronises...?

Even within classical process algebras there is variation in the interpretation of parallel composition:

CCS-style

- Actions are partitioned into input and output pairs.

Who Synchronises...?

Even within classical process algebras there is variation in the interpretation of parallel composition:

CCS-style

- Actions are partitioned into input and output pairs.
- Communication or synchronisation takes places between conjugate pairs.

Who Synchronises...?

Even within classical process algebras there is variation in the interpretation of parallel composition:

CCS-style

- Actions are partitioned into input and output pairs.
- Communication or synchronisation takes places between conjugate pairs.
- The resulting action has silent type τ.

Who Synchronises...?

Even within classical process algebras there is variation in the interpretation of parallel composition:

CCS-style

- Actions are partitioned into input and output pairs.
- Communication or synchronisation takes places between conjugate pairs.
- The resulting action has silent type τ.

CSP-style

- No distinction between input and output actions.

Who Synchronises...?

Even within classical process algebras there is variation in the interpretation of parallel composition:

CCS-style

- Actions are partitioned into input and output pairs.
- Communication or synchronisation takes places between conjugate pairs.
- The resulting action has silent type τ.

CSP-style

- No distinction between input and output actions.
- Communication or synchronisation takes place on the basis of shared names.

Who Synchronises...?

Even within classical process algebras there is variation in the interpretation of parallel composition:

CCS-style

- Actions are partitioned into input and output pairs.
- Communication or synchronisation takes places between conjugate pairs.
- The resulting action has silent type τ.

CSP-style

- No distinction between input and output actions.
- Communication or synchronisation takes place on the basis of shared names.
- The resulting action has the same name.

Who Synchronises...?

Even within classical process algebras there is variation in the interpretation of parallel composition:

CCS-style

- Actions are partitioned into input and output pairs.
- Communication or synchronisation takes places between conjugate pairs.
- The resulting action has silent type τ.

CSP-style

- No distinction between input and output actions.
- Communication or synchronisation takes place on the basis of shared names.
- The resulting action has the same name.

We will see examples of both CCS-style and CSP-style synchronisation.

Timed Synchronisation

- The issue of what it means for two timed activities to synchronise is a vexed one....

Timed Synchronisation

- The issue of what it means for two timed activities to synchronise is a vexed one....
- In a computing context different components may have different capacities to carry out an activity.

Timed Synchronisation

- The issue of what it means for two timed activities to synchronise is a vexed one....
- In a computing context different components may have different capacities to carry out an activity.
- The rate of a synchronised or shared activity must then be chosen, reflecting the capacities of the components involved.

Timed Synchronisation

- The issue of what it means for two timed activities to synchronise is a vexed one....
- In a computing context different components may have different capacities to carry out an activity.
- The rate of a synchronised or shared activity must then be chosen, reflecting the capacities of the components involved.
- The different SPA languages have adopted a number of different solutions to this problem.

Cooperation in PEPA

- In PEPA each component has a bounded capacity to carry out activities of any particular type, determined by the apparent rate for that type.

Cooperation in PEPA

- In PEPA each component has a bounded capacity to carry out activities of any particular type, determined by the apparent rate for that type.
- Synchronisation, or cooperation cannot make a component exceed its bounded capacity.

Cooperation in PEPA

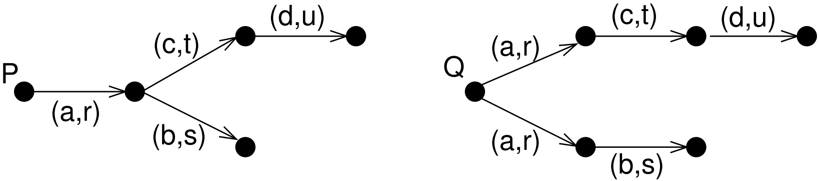
- In PEPA each component has a bounded capacity to carry out activities of any particular type, determined by the apparent rate for that type.
- Synchronisation, or cooperation cannot make a component exceed its bounded capacity.
- Thus the apparent rate of a cooperation is the minimum of the apparent rates of the co-operands.

Cooperation in PEPA

- In PEPA each component has a bounded capacity to carry out activities of any particular type, determined by the apparent rate for that type.
- Synchronisation, or cooperation cannot make a component exceed its bounded capacity.
- Thus the apparent rate of a cooperation is the minimum of the apparent rates of the co-operands.
- We will see that a different solution is appropriate in the context of biological systems.

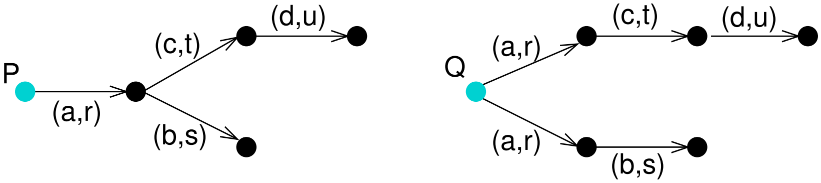
Equivalence Relations

In process algebra equivalence relations are defined based on the notion of observability:



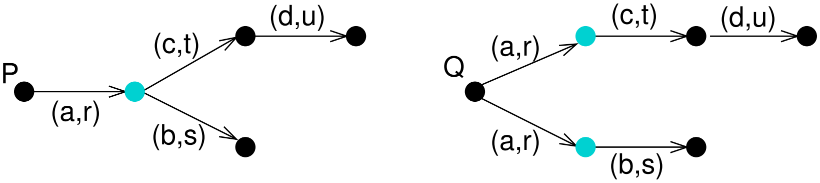
Equivalence Relations

In process algebra equivalence relations are defined based on the notion of observability:



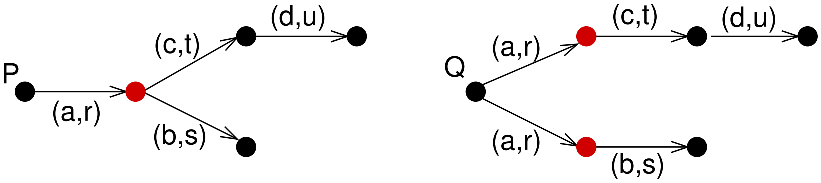
Equivalence Relations

In process algebra equivalence relations are defined based on the notion of observability:



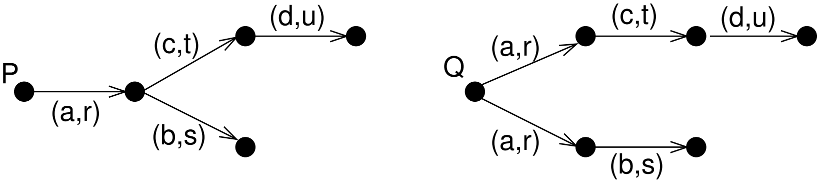
Equivalence Relations

In process algebra equivalence relations are defined based on the notion of observability:



Equivalence Relations

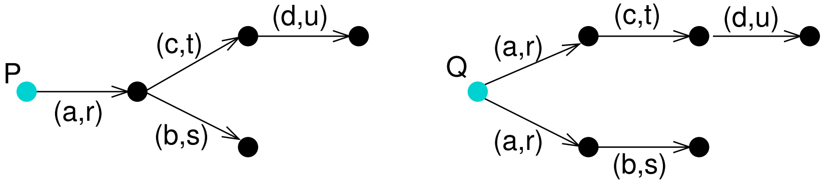
In process algebra equivalence relations are defined based on the notion of observability:



In SPA observation is assumed to include the ability to record timing information over a number of runs.

Equivalence Relations

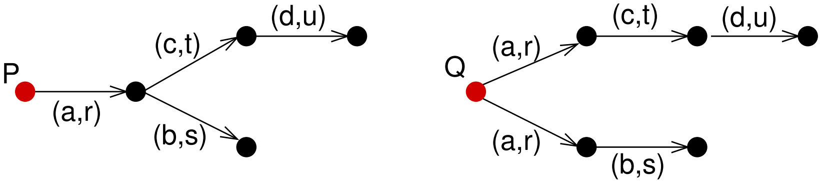
In process algebra equivalence relations are defined based on the notion of observability:



In SPA observation is assumed to include the ability to record timing information over a number of runs.

Equivalence Relations

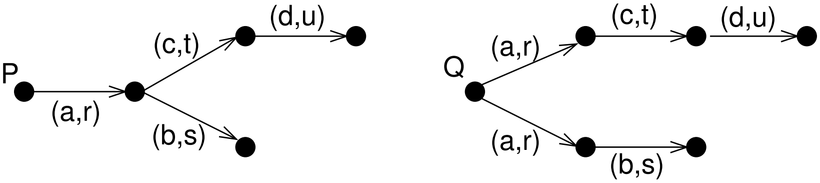
In process algebra equivalence relations are defined based on the notion of observability:



In SPA observation is assumed to include the ability to record timing information over a number of runs.

Equivalence Relations

In process algebra equivalence relations are defined based on the notion of observability:



In SPA observation is assumed to include the ability to record timing information over a number of runs.

The resulting equivalence relation is a bisimulation in the style of Larsen and Skou, and coincides with the Markov process notion of lumpability.

Exploiting equivalence relations

In a SPA model an equivlance relation may be used in two ways to assist model solution:

Exploiting equivalence relations

In a SPA model an equivlance relation may be used in two ways to assist model solution:

- Equivalence between models: The behaviour of two alternative models/components may be compared. Equivalent ones may be used interchangeably. This is of particular value when one model is easier to solve than the other e.g. if it has a smaller state space. This is termed model simplification.

Exploiting equivalence relations

In a SPA model an equivlance relation may be used in two ways to assist model solution:

- Equivalence between models: The behaviour of two alternative models/components may be compared. Equivalent ones may be used interchangeably. This is of particular value when one model is easier to solve than the other e.g. if it has a smaller state space. This is termed model simplification.
- Equivalence within a model: The behaviour of individual states within the state space of a single model may be compared. This can lead to the formation of equivalence classes and a more abstract representation may then be chosen with one representative of each equivalence class. This is termed model aggregation.

Aggregation and lumpability

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, i.e. take a different stochastic representation of the same model.

Aggregation and lumpability

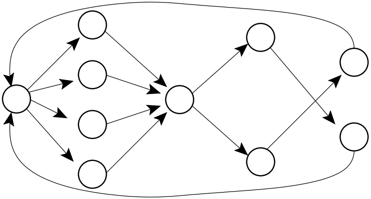
Model aggregation: use a state-state equivalence to establish a partition of the state space of a model, and replace each set of states by one macro-state, i.e. take a different stochastic representation of the same model.

A lumpable partition is the only partition of a Markov process which preserves the Markov property.

Aggregation and lumpability

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, i.e. take a different stochastic representation of the same model.

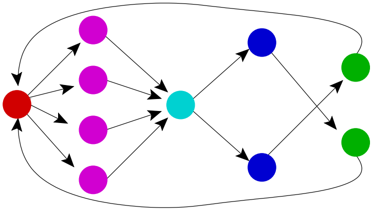
A lumpable partition is the only partition of a Markov process which preserves the Markov property.



Aggregation and lumpability

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, i.e. take a different stochastic representation of the same model.

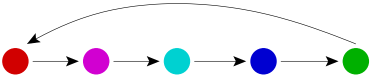
A lumpable partition is the only partition of a Markov process which preserves the Markov property.



Aggregation and lumpability

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, i.e. take a different stochastic representation of the same model.

A lumpable partition is the only partition of a Markov process which preserves the Markov property.

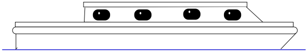


PEPA Case Studies (1)

- Multiprocessor access-contention protocols (Gilmore, Hillston and Ribaudo, Edinburgh and Turin)
- Protocols for fault-tolerant systems (Clark, Gilmore, Hillston and Ribaudo, Edinburgh and Turin)
- Multimedia traffic characteristics (Bowman et al, Kent)
- Database systems (The STEADY group, Heriot-Watt University)
- Software Architectures (Pooley, Bradley and Thomas, Heriot-Watt and Durham)
- Switch behaviour in active networks (Hillston, Kloul and Mokhtari, Edinburgh and Versailles)

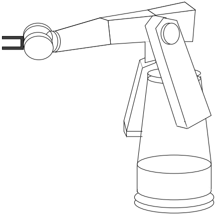
PEPA Case Studies (2)

- Locks and movable bridges in inland shipping in Belgium (Knapen, Hasselt)



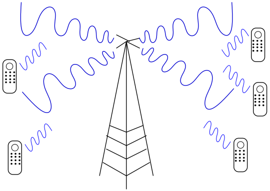
PEPA Case Studies (2)

- Locks and movable bridges in inland shipping in Belgium (Knapen, Hasselt)
- Robotic workcells (Holton, Gilmore and Hillston, Bradford and Edinburgh)



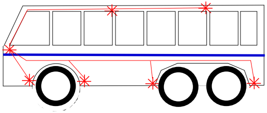
PEPA Case Studies (2)

- Locks and movable bridges in inland shipping in Belgium (Knapen, Hasselt)
- Robotic workcells (Holton, Gilmore and Hillston, Bradford and Edinburgh)
- Cellular telephone networks (Kloul, Fourneau and Valois, Versailles)

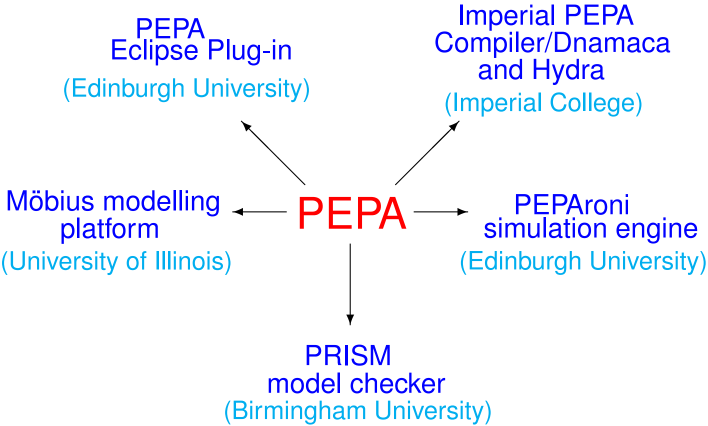


PEPA Case Studies (2)

- Locks and movable bridges in inland shipping in Belgium (Knapen, Hasselt)
- Robotic workcells (Holton, Gilmore and Hillston, Bradford and Edinburgh)
- Cellular telephone networks (Kloul, Fourneau and Valois, Versailles)
- Automotive diagnostic expert systems (Console, Picardi and Ribaudo, Turin)



Tool Support



PEPA

PEPA
Eclipse Plug-in
(Edinburgh University)

Imperial PEPA
Compiler/Dnamaca
and Hydra
(Imperial College)

Möbius modelling
platform
(University of Illinois)

PEPAroni
simulation engine
(Edinburgh University)

PRISM
model checker
(Birmingham University)

Markovian analysis

- Analysis of the Markov process can yield quite detailed information about the dynamic behaviour of the model.

Markovian analysis

- Analysis of the Markov process can yield quite detailed information about the dynamic behaviour of the model.
- A steady state analysis provides statistics for average behaviour over a long run of the system, when the bias introduced by the initial state has been lost.

Markovian analysis

- Analysis of the Markov process can yield quite detailed information about the dynamic behaviour of the model.
- A steady state analysis provides statistics for average behaviour over a long run of the system, when the bias introduced by the initial state has been lost.
- A transient analysis provides statistics relating to the evolution of the model over a fixed period. This will be dependent on the starting state.

Markovian analysis

- Analysis of the Markov process can yield quite detailed information about the dynamic behaviour of the model.
- A steady state analysis provides statistics for average behaviour over a long run of the system, when the bias introduced by the initial state has been lost.
- A transient analysis provides statistics relating to the evolution of the model over a fixed period. This will be dependent on the starting state.
- Stochastic model checking is available via the PRISM model checker, assessing the probable validity of properties expressed in CSL (Continuous Stochastic Logic).

Outline

Process Calculi

SPA

SPA for Systems Biology

Bio-PEPA

Model definition

The syntax and semantics

Equivalences and Analysis

Examples

Genetic network with negative feedback loop

Goldbeter's model

Conclusions

Using Stochastic Process Algebras

Process algebras have several attractive features which could be useful for modelling and understanding biological systems:

Using Stochastic Process Algebras

Process algebras have several attractive features which could be useful for modelling and understanding biological systems:

- Process algebraic formulations are compositional and make interactions/constraints explicit.

Using Stochastic Process Algebras

Process algebras have several attractive features which could be useful for modelling and understanding biological systems:

- Process algebraic formulations are compositional and make interactions/constraints explicit.
- Structure can also be apparent.

Using Stochastic Process Algebras

Process algebras have several attractive features which could be useful for modelling and understanding biological systems:

- Process algebraic formulations are compositional and make interactions/constraints explicit.
- Structure can also be apparent.
- Equivalence relations allow formal comparison of high-level descriptions.

Using Stochastic Process Algebras

Process algebras have several attractive features which could be useful for modelling and understanding biological systems:

- Process algebraic formulations are compositional and make interactions/constraints explicit.
- Structure can also be apparent.
- Equivalence relations allow formal comparison of high-level descriptions.
- There are well-established techniques for reasoning about the behaviours and properties of models, supported by software. These include qualitative and quantitative analysis, and model checking.

Molecular processes as concurrent computations

Concurrency	**Molecular Biology**	**Metabolism**	**Signal Transduction**
Concurrent computational processes	Molecules	Enzymes and metabolites	Interacting proteins
Synchronous communication	Molecular interaction	Binding and catalysis	Binding and catalysis
Transition or mobility	Biochemical modification or relocation	Metabolite synthesis	Protein binding, modification or sequestration

[Regev *et al* 2000]

Molecular processes as concurrent computations

Concurrency	**Molecular Biology**	**Metabolism**	**Signal Transduction**
Concurrent computational processes	Molecules	Enzymes and metabolites	Interacting proteins
Synchronous communication	Molecular interaction	Binding and catalysis	Binding and catalysis
Transition or mobility	Biochemical modification or relocation	Metabolite synthesis	Protein binding, modification or sequestration

[Regev *et al* 2000]

Calculi for Systems Biology

Several different process caluli have been developed or adapted for application in systems biology. Each of them has different properties able to render different aspects of biological phenomena. They may be divided into two main categories:

Calculi for Systems Biology

Several different process caluli have been developed or adapted for application in systems biology. Each of them has different properties able to render different aspects of biological phenomena. They may be divided into two main categories:

- Calculi defined originally in computer science and then applied in biology, such as the biochemical stochastic π-calculus, SCCP, CCS-R and PEPA;

Calculi for Systems Biology

Several different process caluli have been developed or adapted for application in systems biology. Each of them has different properties able to render different aspects of biological phenomena. They may be divided into two main categories:

- Calculi defined originally in computer science and then applied in biology, such as the biochemical stochastic π-calculus, SCCP, CCS-R and PEPA;
- Calculi defined specifically by observing biological structures and phenomena, such as BioAmbients, Brane Calculi and Beta-binders.

Stochastic π-calculus

- The stochastic π-calculus has been used to model and analyse a wide variety of biological systems.

Stochastic π-calculus

- The stochastic π-calculus has been used to model and analyse a wide variety of biological systems.
- Examples include metabolic pathways, gene transcription and signal transduction.

Stochastic π-calculus

- The stochastic π-calculus has been used to model and analyse a wide variety of biological systems.
- Examples include metabolic pathways, gene transcription and signal transduction.
- Analysis is mainly based on stochastic simulation (Gillespie's algorithm).

Stochastic π-calculus

- The stochastic π-calculus has been used to model and analyse a wide variety of biological systems.
- Examples include metabolic pathways, gene transcription and signal transduction.
- Analysis is mainly based on stochastic simulation (Gillespie's algorithm).
- Two tools: BioSPI and SPIM which implement slightly different versions of the language.

Stochastic π-calculus

- The stochastic π-calculus has been used to model and analyse a wide variety of biological systems.
- Examples include metabolic pathways, gene transcription and signal transduction.
- Analysis is mainly based on stochastic simulation (Gillespie's algorithm).
- Two tools: BioSPI and SPIM which implement slightly different versions of the language.
- There has also been some work on a graphical notation associated with the SPIM tool.

CCS-R

- CCS-R is a variant of CCS with new elements which allow the capture of reversibility.

CCS-R

- CCS-R is a variant of CCS with new elements which allow the capture of reversibility.
- The interactions are described in terms of binary synchronised communications, similarly to π-calculus.

CCS-R

- CCS-R is a variant of CCS with new elements which allow the capture of reversibility.
- The interactions are described in terms of binary synchronised communications, similarly to π-calculus.
- It was motivated by modelling reversible reactions in biochemistry.

CCS-R

- CCS-R is a variant of CCS with new elements which allow the capture of reversibility.
- The interactions are described in terms of binary synchronised communications, similarly to π-calculus.
- It was motivated by modelling reversible reactions in biochemistry.
- The successor of CSS-R is the Reversible CCS (RCCS). This calculus allows processes to backtrack if this is in agreement with a defined notion of casual equivalence.

SCCP

- SCCP is a stochastic extension of Concurrent Constraint Programming (CCP).

SCCP

- SCCP is a stochastic extension of Concurrent Constraint Programming (CCP).
- Here each species is represented by a variable and the reactions are expressed by constraints (or updates) on these variables.

SCCP

- SCCP is a stochastic extension of Concurrent Constraint Programming (CCP).
- Here each species is represented by a variable and the reactions are expressed by constraints (or updates) on these variables.
- Mappings have been defined from SCCP models to CTMC models at the molecular level suitable for stochastic simulation, to systems of ODEs and to hybrid systems.

SCCP

- SCCP is a stochastic extension of Concurrent Constraint Programming (CCP).
- Here each species is represented by a variable and the reactions are expressed by constraints (or updates) on these variables.
- Mappings have been defined from SCCP models to CTMC models at the molecular level suitable for stochastic simulation, to systems of ODEs and to hybrid systems.
- Unlike the stochastic π-calculus and CCS-R, the calculus is not limited to binary interactions so any kind of reactions can be represented.

SCCP

- SCCP is a stochastic extension of Concurrent Constraint Programming (CCP).
- Here each species is represented by a variable and the reactions are expressed by constraints (or updates) on these variables.
- Mappings have been defined from SCCP models to CTMC models at the molecular level suitable for stochastic simulation, to systems of ODEs and to hybrid systems.
- Unlike the stochastic π-calculus and CCS-R, the calculus is not limited to binary interactions so any kind of reactions can be represented.
- Moreover the rate can be expressed by a generic function, thus general kinetic laws can be captured.

Biology-specific process calculi

The current work on defining biology-specific process calculi has focused on spatial aspects.

Biology-specific process calculi

The current work on defining biology-specific process calculi has focused on spatial aspects.

Thus each of the new calculi places emphasis on the location of components and how this impacts on their potential interactions.

The Bioambient Calculus

- Originating from Regev *et al.*

The Bioambient Calculus

- Originating from Regev *et al.*
- An extension of the stochastic π-calculus, inspired by the Ambient Calculus which captures process location and movement through computational domains.

The Bioambient Calculus

- Originating from Regev *et al.*
- An extension of the stochastic π-calculus, inspired by the Ambient Calculus which captures process location and movement through computational domains.
- An ambient is a bounded place where computation can happen — within each ambient there may be component processes and sub-ambients.

The Bioambient Calculus

- Originating from Regev *et al.*
- An extension of the stochastic π-calculus, inspired by the Ambient Calculus which captures process location and movement through computational domains.
- An ambient is a bounded place where computation can happen — within each ambient there may be component processes and sub-ambients.
- Entities may enter or exit an ambient and ambients may merge.

The Bioambient Calculus

- Originating from Regev *et al.*
- An extension of the stochastic π-calculus, inspired by the Ambient Calculus which captures process location and movement through computational domains.
- An ambient is a bounded place where computation can happen — within each ambient there may be component processes and sub-ambients.
- Entities may enter or exit an ambient and ambients may merge.
- A stochastic version has recently been defined and used in applications.

The Brane Calculus

- Originating from Cardelli at Microsoft Research, Cambridge.

The Brane Calculus

- Originating from Cardelli at Microsoft Research, Cambridge.
- Membranes are taken as primitives of the calculus:

The Brane Calculus

- Originating from Cardelli at Microsoft Research, Cambridge.
- Membranes are taken as primitives of the calculus:
  - Systems consist of nested membranes;

The Brane Calculus

- Originating from Cardelli at Microsoft Research, Cambridge.
- Membranes are taken as primitives of the calculus:
  - Systems consist of nested membranes;
  - Membranes consist of collections of actions;

The Brane Calculus

- Originating from Cardelli at Microsoft Research, Cambridge.
- Membranes are taken as primitives of the calculus:
  - Systems consist of nested membranes;
  - Membranes consist of collections of actions;
  - Reactions happen only at the level of systems and are caused only by actions on membranes.

The Brane Calculus

- Originating from Cardelli at Microsoft Research, Cambridge.
- Membranes are taken as primitives of the calculus:
  - Systems consist of nested membranes;
  - Membranes consist of collections of actions;
  - Reactions happen only at the level of systems and are caused only by actions on membranes.
  - Actions may be bitonal actions of the membrane, binding or release, or molecular interactions.

Beta Binders

- Originating from the group at the University of Trento.

Beta Binders

- Originating from the group at the University of Trento.
- Can be viewed as an extension of the stochastic π-calculus.

Beta Binders

- Originating from the group at the University of Trento.
- Can be viewed as an extension of the stochastic π-calculus.
- Processes are encapsulated into boxes, that mimic biological membranes, and have interaction sites, motifs, depicting where molecules can bind together.

Beta Binders

- Originating from the group at the University of Trento.
- Can be viewed as an extension of the stochastic π-calculus.
- Processes are encapsulated into boxes, that mimic biological membranes, and have interaction sites, motifs, depicting where molecules can bind together.
- The semantics give rules on joining and splitting boxes, as well as the affinity between interaction sites.

Mapping biological systems to process algebra

The work using the stochastic π-calculus and related calculi, maps a molecule to a process in the process algebra description.

Mapping biological systems to process algebra

The work using the stochastic π-calculus and related calculi, maps a molecule to a process in the process algebra description.

This is an inherently individuals-based view of the system and analysis will generally be via stochastic simulation.

Mapping biological systems to process algebra

The work using the stochastic π-calculus and related calculi, maps a molecule to a process in the process algebra description.

This is an inherently individuals-based view of the system and analysis will generally be via stochastic simulation.

In the PEPA group we have been experimenting with more abstract mappings between process algebra constructs and elements of signalling pathways.

Mapping biological systems to process algebra

The work using the stochastic π-calculus and related calculi, maps a molecule to a process in the process algebra description.

This is an inherently individuals-based view of the system and analysis will generally be via stochastic simulation.

In the PEPA group we have been experimenting with more abstract mappings between process algebra constructs and elements of signalling pathways.

In our mapping we focus on species (c.f. a type rather than an instance, or a class rather than an object).

Mapping biological systems to process algebra

The work using the stochastic π-calculus and related calculi, maps a molecule to a process in the process algebra description.

This is an inherently individuals-based view of the system and analysis will generally be via stochastic simulation.

In the PEPA group we have been experimenting with more abstract mappings between process algebra constructs and elements of signalling pathways.

In our mapping we focus on species (c.f. a type rather than an instance, or a class rather than an object).

Alternative mappings from the process algebra to underlying mathematics are then readily available.

Motivations for Abstraction

Our motivations for seeking more abstraction in process algebra models for systems biology are:

Motivations for Abstraction

Our motivations for seeking more abstraction in process algebra models for systems biology are:

- Process algebra-based analyses such as comparing models (e.g. for equivalence or simulation) and model checking are only possible is the state space is not prohibitively large.

Motivations for Abstraction

Our motivations for seeking more abstraction in process algebra models for systems biology are:

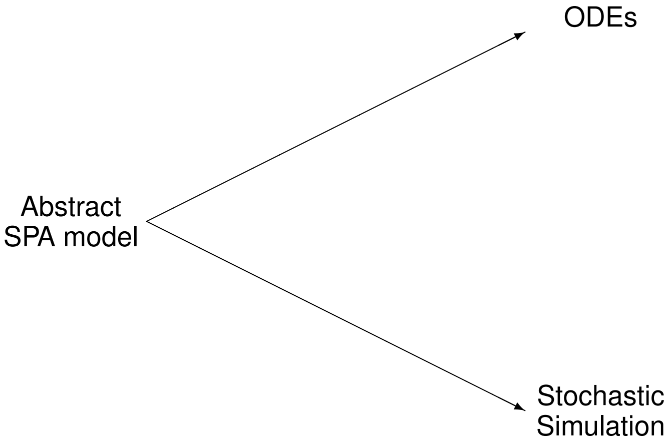
- Process algebra-based analyses such as comparing models (e.g. for equivalence or simulation) and model checking are only possible is the state space is not prohibitively large.
- The data that we have available to parameterise models is sometimes speculative rather than precise.

Motivations for Abstraction

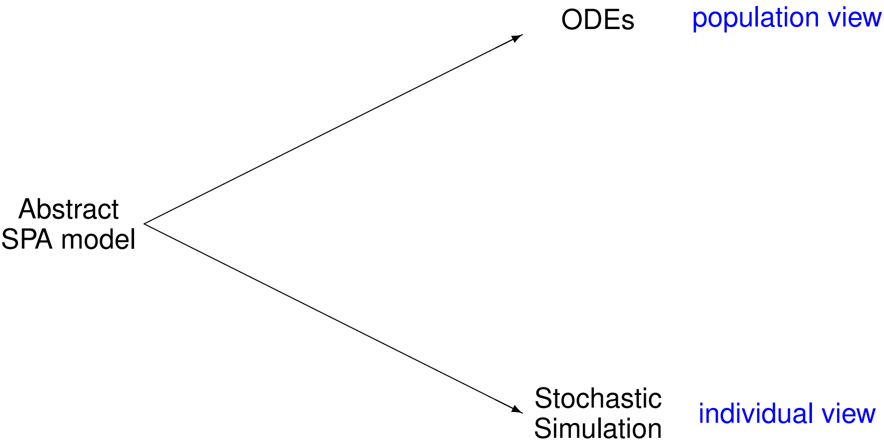
Our motivations for seeking more abstraction in process algebra models for systems biology are:

- Process algebra-based analyses such as comparing models (e.g. for equivalence or simulation) and model checking are only possible is the state space is not prohibitively large.
- The data that we have available to parameterise models is sometimes speculative rather than precise. This suggests that it can be useful to use semiquantitative models rather than quantitative ones.

Alternative Representations



Alternative Representations



Abstract SPA model

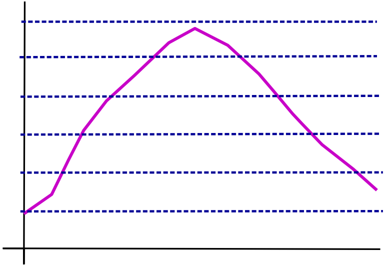
ODEs

population view

Stochastic Simulation

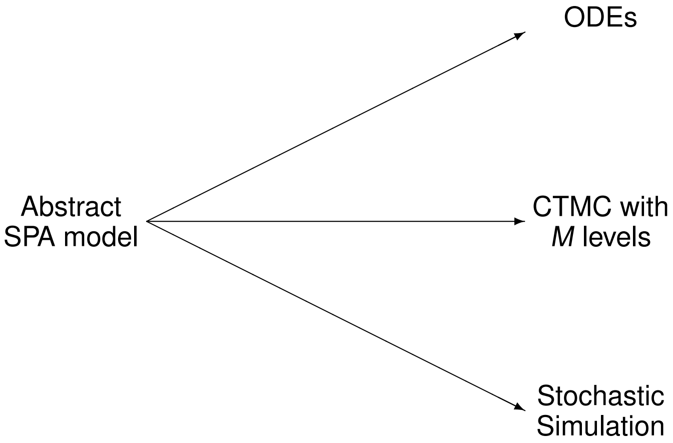
individual view

Discretising the population view



We can discretise the continuous range of possible concentration values into a number of distinct states. These form the possible states of the component representing the reagent.

Alternative Representations



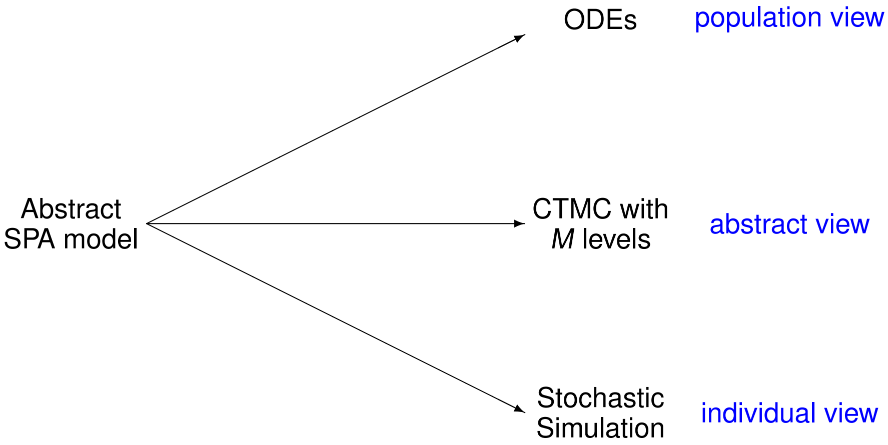
Abstract SPA model

ODEs

CTMC with *M* levels

Stochastic Simulation

Alternative Representations



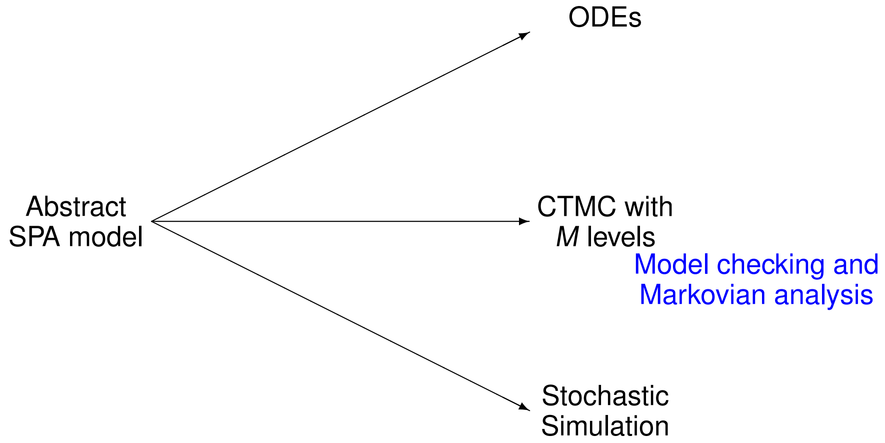
Abstract SPA model

ODEs population view

CTMC with *M* levels abstract view

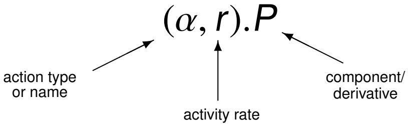
Stochastic Simulation individual view

Alternative Representations



Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$$(\alpha, r).P$$

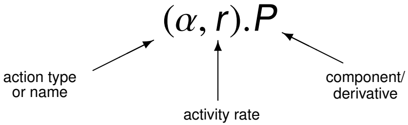
action type
or name

activity rate

component/
derivative

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$$(\alpha, r).P$$

action type or name

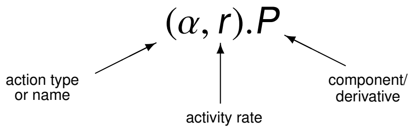
activity rate

component/ derivative

The language may be used to generate a Markov Process (CTMC).

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$$(\alpha, r).P$$

action type or name

activity rate

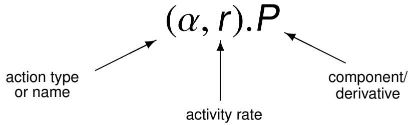
component/ derivative

The language may be used to generate a Markov Process (CTMC).

SPA
MODEL

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



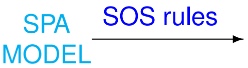
$(\alpha, r).P$

action type or name

activity rate

component/ derivative

The language may be used to generate a Markov Process (CTMC).

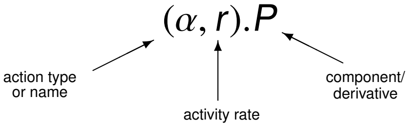


SPA MODEL

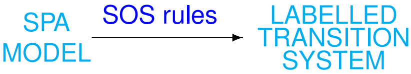
SOS rules

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

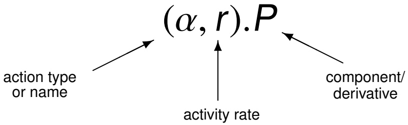


The language may be used to generate a Markov Process (CTMC).



Stochastic Process Algebra

- Models are constructed from components which engage in activities.



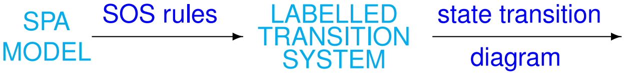
$(\alpha, r).P$

action type or name

activity rate

component/ derivative

The language may be used to generate a Markov Process (CTMC).



SPA MODEL

SOS rules

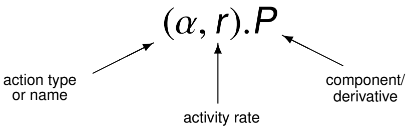
LABELLED TRANSITION SYSTEM

state transition

diagram

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$(\alpha, r).P$

action type or name

activity rate

component/ derivative

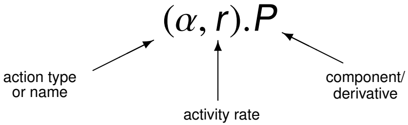
The language may be used to generate a Markov Process (CTMC).

$$\text{SPA MODEL} \xrightarrow{\text{SOS rules}} \text{LABELLED TRANSITION SYSTEM} \xrightarrow[\text{diagram}]{\text{state transition}} \text{CTMC } \mathbf{Q}$$

Q is the infinitesimal generator matrix characterising the CTMC.

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$$(\alpha, r).P$$

action type or name

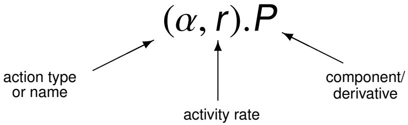
activity rate

component/ derivative

The language may be used to generate a system of ordinary differential equations (ODEs).

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



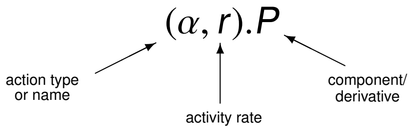
$$(\alpha, r).P$$

The language may be used to generate a system of ordinary differential equations (ODEs).

SPA
MODEL

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



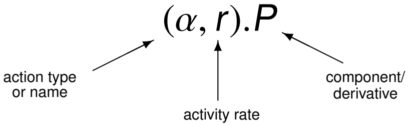
$(\alpha, r).P$

The language may be used to generate a system of ordinary differential equations (ODEs).

$$\text{SPA MODEL} \xrightarrow[\text{analysis}]{\text{syntactic}}$$

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$(\alpha, r).P$

action type or name

activity rate

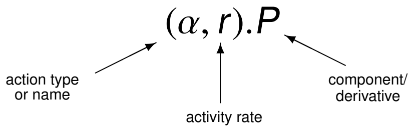
component/ derivative

The language may be used to generate a system of ordinary differential equations (ODEs).

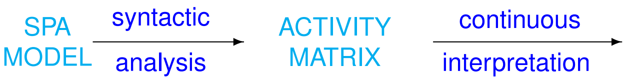
$$\text{SPA MODEL} \xrightarrow[\text{analysis}]{\text{syntactic}} \text{ACTIVITY MATRIX}$$

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

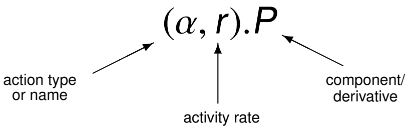


The language may be used to generate a system of ordinary differential equations (ODEs).



Stochastic Process Algebra

- Models are constructed from components which engage in activities.



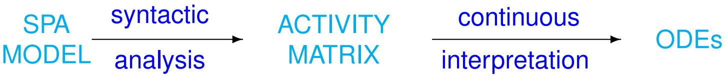
$(\alpha, r).P$

action type or name

activity rate

component/ derivative

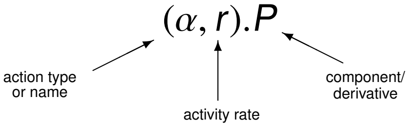
The language may be used to generate a system of ordinary differential equations (ODEs).



SPA MODEL $\xrightarrow[\text{analysis}]{\text{syntactic}}$ ACTIVITY MATRIX $\xrightarrow[\text{interpretation}]{\text{continuous}}$ ODEs

Stochastic Process Algebra

- Models are constructed from components which engage in activities.

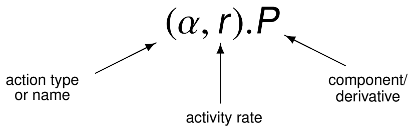


$$(\alpha, r).P$$

The language also may be used to generate a stochastic simulation.

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$$(\alpha, r).P$$

action type or name

activity rate

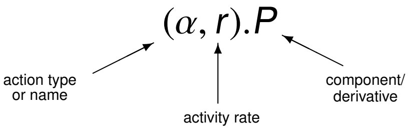
component/ derivative

The language also may be used to generate a stochastic simulation.

SPA
MODEL

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$$(\alpha, r).P$$

action type or name

activity rate

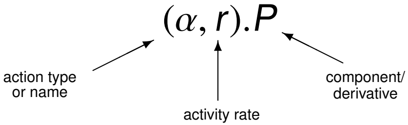
component/ derivative

The language also may be used to generate a stochastic simulation.

$$\text{SPA MODEL} \xrightarrow[\text{analysis}]{\text{syntactic}}$$

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$(\alpha, r).P$

action type or name

activity rate

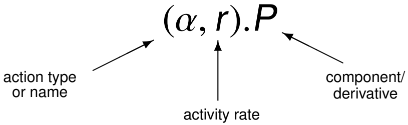
component/ derivative

The language also may be used to generate a stochastic simulation.

$$\text{SPA MODEL} \xrightarrow[\text{analysis}]{\text{syntactic}} \text{RATE EQUATIONS}$$

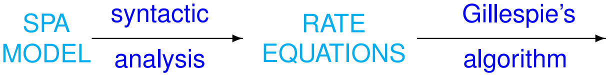
Stochastic Process Algebra

- Models are constructed from components which engage in activities.



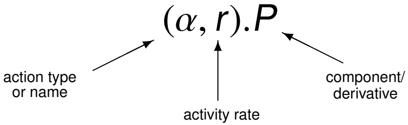
$$(\alpha, r).P$$

The language also may be used to generate a stochastic simulation.



Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$(\alpha, r).P$

action type or name

activity rate

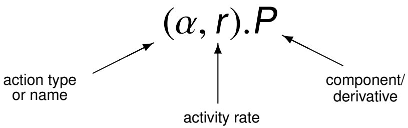
component/ derivative

The language also may be used to generate a stochastic simulation.

$$\text{SPA MODEL} \xrightarrow[\text{analysis}]{\text{syntactic}} \text{RATE EQUATIONS} \xrightarrow[\text{algorithm}]{\text{Gillespie's}} \text{STOCHASTIC SIMULATION}$$

Stochastic Process Algebra

- Models are constructed from components which engage in activities.



$$(\alpha, r).P$$

The language also may be used to generate a stochastic simulation.

$$\text{SPA MODEL} \xrightarrow[\text{analysis}]{\text{syntactic}} \text{RATE EQUATIONS} \xrightarrow[\text{algorithm}]{\text{Gillespie's}} \text{STOCHASTIC SIMULATION}$$

Alternative models

- When a molecular mapping is used in general a CTMC state space is too large to permit anything but stochastic simulation.

Alternative models

- When a molecular mapping is used in general a CTMC state space is too large to permit anything but stochastic simulation.
- The ODE model can be regarded as an approximation of a CTMC in which the number of molecules is large enough that the randomness averages out and the system is essentially deterministic.

Alternative models

- When a molecular mapping is used in general a CTMC state space is too large to permit anything but stochastic simulation.
- The ODE model can be regarded as an approximation of a CTMC in which the number of molecules is large enough that the randomness averages out and the system is essentially deterministic.
- In models with levels, each level of granularity gives rise to a CTMC, and the behaviour of this sequence of Markov processes converges to the behaviour of the system of ODEs.

Alternative models

- When a molecular mapping is used in general a CTMC state space is too large to permit anything but stochastic simulation.
- The ODE model can be regarded as an approximation of a CTMC in which the number of molecules is large enough that the randomness averages out and the system is essentially deterministic.
- In models with levels, each level of granularity gives rise to a CTMC, and the behaviour of this sequence of Markov processes converges to the behaviour of the system of ODEs.
- Some analyses which can be carried out via numerical solution of the CTMC are not readily available from ODEs or stochastic simulation.

Outline

Process Calculi

SPA

SPA for Systems Biology

Bio-PEPA

Model definition

The syntax and semantics

Equivalences and Analysis

Examples

Genetic network with negative feedback loop

Goldbeter's model

Conclusions

Modelling biological features

SPA designed for modelling computing systems do not readily capture some of the features of biological systems.

Modelling biological features

SPA designed for modelling computing systems do not readily capture some of the features of biological systems.

Particular problems are encountered with:

Modelling biological features

SPA designed for modelling computing systems do not readily capture some of the features of biological systems.

Particular problems are encountered with:

- stoichiometry — the multiplicity in which an entity participates in a reaction;

Modelling biological features

SPA designed for modelling computing systems do not readily capture some of the features of biological systems.

Particular problems are encountered with:

- stoichiometry — the multiplicity in which an entity participates in a reaction;
- general kinetic laws — while mass action is widely used other kinetics are also commonly employed.

Modelling biological features

SPA designed for modelling computing systems do not readily capture some of the features of biological systems.

Particular problems are encountered with:

- stoichiometry — the multiplicity in which an entity participates in a reaction;
- general kinetic laws — while mass action is widely used other kinetics are also commonly employed.
- multiway reactions — although thermodynamics arguments can be made that there are never more than two reagents involved in a reaction, in practice it is often useful to model at a more abstract level.

Illustration

Consider a conversion of a substrate S, with stoichiometry 2, to a product P which is under the influence of an enzyme E, i.e.

$$2S \xrightarrow{E} P$$

Illustration

Consider a conversion of a substrate S, with stoichiometry 2, to a product P which is under the influence of an enzyme E, i.e.

$$2S \xrightarrow{E} P$$

In process algebras such as the stochastic π-calculus this must be broken up into a sequence of unary and binary reactions, e.g.:

$$S + S \longrightarrow 2S \quad 2S + E \longrightarrow 2S{:}E \quad 2S{:}E \longrightarrow P{:}E \quad P{:}E \longrightarrow P + E$$

Illustration

Consider a conversion of a substrate S, with stoichiometry 2, to a product P which is under the influence of an enzyme E, i.e.

$$2S \xrightarrow{E} P$$

In process algebras such as the stochastic π-calculus this must be broken up into a sequence of unary and binary reactions, e.g.:

$$\color{red}{S + S \longrightarrow 2S} \quad 2S + E \longrightarrow 2S{:}E \quad 2S{:}E \longrightarrow P{:}E \quad P{:}E \longrightarrow P + E$$

Illustration

Consider a conversion of a substrate S, with stoichiometry 2, to a product P which is under the influence of an enzyme E, i.e.

$$2S \xrightarrow{E} P$$

In process algebras such as the stochastic π-calculus this must be broken up into a sequence of unary and binary reactions, e.g.:

$$S + S \longrightarrow 2S \quad 2S + E \longrightarrow 2S : E \quad 2S : E \longrightarrow P : E \quad P : E \longrightarrow P + E$$

Illustration

Consider a conversion of a substrate S, with stoichiometry 2, to a product P which is under the influence of an enzyme E, i.e.

$$2S \xrightarrow{E} P$$

In process algebras such as the stochastic π-calculus this must be broken up into a sequence of unary and binary reactions, e.g.:

$$S + S \longrightarrow 2S \quad 2S + E \longrightarrow 2S\!:\!E \quad 2S\!:\!E \longrightarrow P\!:\!E \quad P\!:\!E \longrightarrow P + E$$

Illustration

Consider a conversion of a substrate S, with stoichiometry 2, to a product P which is under the influence of an enzyme E, i.e.

$$2S \xrightarrow{E} P$$

In process algebras such as the stochastic π-calculus this must be broken up into a sequence of unary and binary reactions, e.g.:

$$S + S \longrightarrow 2S \quad 2S + E \longrightarrow 2S:E \quad 2S:E \longrightarrow P:E \quad P:E \longrightarrow P + E$$

Illustration

Consider a conversion of a substrate S, with stoichiometry 2, to a product P which is under the influence of an enzyme E, i.e.

$$2S \xrightarrow{E} P$$

In process algebras such as the stochastic π-calculus this must be broken up into a sequence of unary and binary reactions, e.g.:

$$S + S \longrightarrow 2S \quad 2S + E \longrightarrow 2S\!:\!E \quad 2S\!:\!E \longrightarrow P\!:\!E \quad P\!:\!E \longrightarrow P + E$$

The problems with this are various:

Illustration

Consider a conversion of a substrate S, with stoichiometry 2, to a product P which is under the influence of an enzyme E, i.e.

$$2S \xrightarrow{E} P$$

In process algebras such as the stochastic π-calculus this must be broken up into a sequence of unary and binary reactions, e.g.:

$$S + S \longrightarrow 2S \quad 2S + E \longrightarrow 2S:E \quad 2S:E \longrightarrow P:E \quad P:E \longrightarrow P + E$$

The problems with this are various:

- The number of “states” of the system is significantly increased which has implications for computational efficiency/tractability.

Illustration

Consider a conversion of a substrate S, with stoichiometry 2, to a product P which is under the influence of an enzyme E, i.e.

$$2S \xrightarrow{E} P$$

In process algebras such as the stochastic π-calculus this must be broken up into a sequence of unary and binary reactions, e.g.:

$$S + S \longrightarrow 2S \quad 2S + E \longrightarrow 2S\!:\!E \quad 2S\!:\!E \longrightarrow P\!:\!E \quad P\!:\!E \longrightarrow P + E$$

The problems with this are various:

- The number of “states” of the system is significantly increased which has implications for computational efficiency/tractability.
- Different possible decompositions.

Illustration

Consider a conversion of a substrate S, with stoichiometry 2, to a product P which is under the influence of an enzyme E, i.e.

$$2S \xrightarrow{E} P$$

In process algebras such as the stochastic π-calculus this must be broken up into a sequence of unary and binary reactions, e.g.:

$$S + S \longrightarrow 2S \quad 2S + E \longrightarrow 2S{:}E \quad 2S{:}E \longrightarrow P{:}E \quad P{:}E \longrightarrow P + E$$

The problems with this are various:

- The number of “states” of the system is significantly increased which has implications for computational efficiency/tractability.
- Different possible decompositions.
- Rates must be found for all the intermediate steps.

Motivation

Bio-PEPA has been designed to overcome these challenges:

Motivation

Bio-PEPA has been designed to overcome these challenges:

- Unique rates are associated with each reaction (action) type, separately from the specification of the logical behaviour. These rates may be specified by functions.

Motivation

Bio-PEPA has been designed to overcome these challenges:

- Unique rates are associated with each reaction (action) type, separately from the specification of the logical behaviour. These rates may be specified by functions.
- The representation of an action within a component (species) records the stoichiometry of that entity with respect to that reaction. The role of the entity is also distinguished.

Motivation

Bio-PEPA has been designed to overcome these challenges:

- Unique rates are associated with each reaction (action) type, separately from the specification of the logical behaviour. These rates may be specified by functions.
- The representation of an action within a component (species) records the stoichiometry of that entity with respect to that reaction. The role of the entity is also distinguished.
- Multi-way reactions are possible in Bio-PEPA since it has CSP-style synchronisation rather than CCS-style synchronisation. Thus a multi-way reaction is abstracted as a multi-syncronisation.

Reagent-centric view [CGH04]

- Bio-PEPA refers to the reagent-centric view modelling style.

Reagent-centric view [CGH04]

- Bio-PEPA refers to the reagent-centric view modelling style.
- Models are based on discrete levels of concentration within a species.

Reagent-centric view [CGH04]

- Bio-PEPA refers to the reagent-centric view modelling style.
- Models are based on discrete levels of concentration within a species.
- The granularity of the system is defined in terms of the step size *h* of the concentration intervals.

Reagent-centric view [CGH04]

- Bio-PEPA refers to the reagent-centric view modelling style.
- Models are based on discrete levels of concentration within a species.
- The granularity of the system is defined in terms of the step size h of the concentration intervals.
- We define the same step size h for all the species, with few exceptions. This follows from the law of conservation of mass.

Reagent-centric view [CGH04]

- Bio-PEPA refers to the reagent-centric view modelling style.
- Models are based on discrete levels of concentration within a species.
- The granularity of the system is defined in terms of the step size *h* of the concentration intervals.
- We define the same step size *h* for all the species, with few exceptions. This follows from the law of conservation of mass.
- If l_i is the concentration level for the species *i*, the concentration is taken to be $x_i = l_i \times h$.

Reagent-centric modelling (2)

Role	*Impact on reaction rate*	*Impact on reagent*
Reactant	positive impact, e.g. proportional to current concentration	decreases level
Product	no impact, except at saturation	increases level
Enzyme	positive impact, e.g. proportional to current concentration	level unchanged
Inhibitor	negative impact, e.g. inversely proportional to current concentration	level unchanged

Reagent-centric view (3)

- The rate of a transition is consistent with the granularity.

Reagent-centric view (3)

- The rate of a transition is consistent with the granularity.
- In a Bio-PEPA model the granularity must be specified by the modeller as the expected range of concentration values and the number of levels considered.

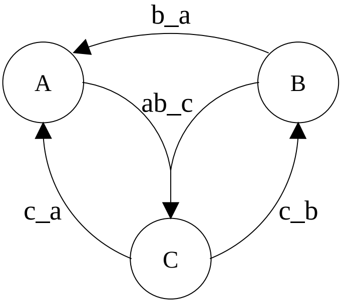
Reagent-centric view (3)

- The rate of a transition is consistent with the granularity.
- In a Bio-PEPA model the granularity must be specified by the modeller as the expected range of concentration values and the number of levels considered.
- The form of the CTMC derived from Bio-PEPA, which we term the CTMC with levels, will depend on the granularity of the model.

Reagent-centric view (3)

- The rate of a transition is consistent with the granularity.
- In a Bio-PEPA model the granularity must be specified by the modeller as the expected range of concentration values and the number of levels considered.
- The form of the CTMC derived from Bio-PEPA, which we term the CTMC with levels, will depend on the granularity of the model.
- As the granularity tends to zero the behaviour of this CTMC with levels tends to the behaviour of the ODEs [CDHC08].

Bio-PEPA reagent-centric example



$$
\begin{aligned}
A &\overset{def}{=} (ab\_c,1)\downarrow A + (b\_a,1)\uparrow A \\
&\quad + (c\_a,1)\uparrow A \\
B &\overset{def}{=} (ab\_c,1)\downarrow B + (b\_a,1)\downarrow B \\
&\quad + (c\_b,1)\uparrow B \\
C &\overset{def}{=} (c\_a,1)\downarrow C + (c\_b,1)\downarrow C \\
&\quad + (ab\_c,1)\uparrow C
\end{aligned}
$$

$$
\left(A(l_{A0}) \underset{\{ab\_c,b\_a\}}{\bowtie} B(l_{B0})\right) \underset{\{ab\_c,c\_a,c\_b\}}{\bowtie} C(l_{C0})
$$

State representation

- The state of the system at any time consists of the local states of each of its sequential/species components.

State representation

- The state of the system at any time consists of the local states of each of its sequential/species components.
- The local states of components are quantitative rather than functional, i.e. distinct states of the species are represented as distinct components, not derivatives of a single component.

State representation

- The state of the system at any time consists of the local states of each of its sequential/species components.
- The local states of components are quantitative rather than functional, i.e. distinct states of the species are represented as distinct components, not derivatives of a single component.
- A component varying its state corresponds to it varying its concentration level.

State representation

- The state of the system at any time consists of the local states of each of its sequential/species components.
- The local states of components are quantitative rather than functional, i.e. distinct states of the species are represented as distinct components, not derivatives of a single component.
- A component varying its state corresponds to it varying its concentration level.
- This is captured by an integer parameter associated with the species and the effect of a reaction is to vary that parameter by a number of levels corresponding to the stoichiometry of this species in the reaction.

The syntax

Sequential (species) component

The syntax

Sequential (species) component

$$S \overset{\mathit{def}}{=} (\alpha, \kappa) \text{ op } S \mid S + S \mid C \qquad \text{where op} = \downarrow \mid \uparrow \mid \oplus \mid \ominus \mid \odot$$

The syntax

Sequential (species) component

$$S \overset{def}{=} (\alpha, \kappa) \text{ op } S \mid S + S \mid C \qquad \text{where op} = \downarrow \mid \uparrow \mid \oplus \mid \ominus \mid \odot$$

Model component

The syntax

Sequential (species) component

$$S \overset{def}{=} (\alpha, \kappa) \text{ op } S \mid S + S \mid C \qquad \text{where op} = \downarrow \mid \uparrow \mid \oplus \mid \ominus \mid \odot$$

Model component

$$P \overset{def}{=} P \underset{\mathcal{L}}{\bowtie} P \mid S(l)$$

The Bio-PEPA system

A Bio-PEPA system $\mathcal{P}$ is a 6-tuple $\langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, Comp, P \rangle$, where:

- $\mathcal{V}$ is the set of compartments;
- $\mathcal{N}$ is the set of quantities describing each species (step size, number of levels, location, ...);
- $\mathcal{K}$ is the set of parameter definitions;
- $\mathcal{F}_R$ is the set of functional rate definitions;
- *Comp* is the set of definitions of sequential components;
- P is the model component describing the system.

Semantics

Semantics

The semantics of Bio-PEPA is defined in terms of an *operational semantics*.

Semantics

The semantics of Bio-PEPA is defined in terms of an *operational semantics*.
We define two relations over the processes:

1. capability relation, that supports the derivation of quantitative information;

Semantics

The semantics of Bio-PEPA is defined in terms of an *operational semantics*.
We define two relations over the processes:

1. capability relation, that supports the derivation of quantitative information;
2. stochastic relation, that gives us the rates associated with each action.

Semantics: prefix rules

Semantics: prefix rules

$$\texttt{prefixReac} \qquad ((\alpha,\kappa){\downarrow}S)(l) \xrightarrow{(\alpha,[S:\downarrow(l,\kappa)])}_{c} S(l-\kappa) \quad \kappa \leq l \leq N$$

Semantics: prefix rules

$$\texttt{prefixReac} \qquad ((\alpha,\kappa){\downarrow}S)(l) \xrightarrow{(\alpha,[S:\downarrow(l,\kappa)])}_c S(l-\kappa) \quad \kappa \leq l \leq N$$

$$\texttt{prefixProd} \qquad ((\alpha,\kappa){\uparrow}S)(l) \xrightarrow{(\alpha,[S:\uparrow(l,\kappa)])}_c S(l+\kappa) \quad 0 \leq l \leq (N-\kappa)$$

Semantics: prefix rules

$$\texttt{prefixReac} \qquad ((\alpha,\kappa){\downarrow}S)(l) \xrightarrow{(\alpha,[S:\downarrow(l,\kappa)])}_c S(l-\kappa) \quad \kappa \le l \le N$$

$$\texttt{prefixProd} \qquad ((\alpha,\kappa){\uparrow}S)(l) \xrightarrow{(\alpha,[S:\uparrow(l,\kappa)])}_c S(l+\kappa) \quad 0 \le l \le (N-\kappa)$$

$$\texttt{prefixMod} \qquad ((\alpha,\kappa)\; op\; S)(l) \xrightarrow{(\alpha,[S:op(l,\kappa)])}_c S(l) \quad 0 \le l \le N$$

with $op = \odot, \oplus$, or $\ominus$

Semantics: constant and choice rules

Semantics: constant and choice rules

$$\texttt{Choice1} \qquad \frac{S_1(l) \xrightarrow{(\alpha,v)}_c S'_1(l')}{(S_1 + S_2)(l) \xrightarrow{(\alpha,v)}_c S'_1(l')}$$

Semantics: constant and choice rules

$$\texttt{Choice1} \quad \frac{S_1(l) \xrightarrow{(\alpha,v)}_c S'_1(l')}{(S_1 + S_2)(l) \xrightarrow{(\alpha,v)}_c S'_1(l')}$$

$$\texttt{Choice2} \quad \frac{S_2(l) \xrightarrow{(\alpha,v)}_c S'_2(l')}{(S_1 + S_2)(l) \xrightarrow{(\alpha,v)}_c S'_2(l')}$$

Semantics: constant and choice rules

Choice1 $$\frac{S_1(l)\xrightarrow{(\alpha,v)}_c S^{'}{}_1(l')}{(S_1 + S_2)(l)\xrightarrow{(\alpha,v)}_c S^{'}{}_1(l')}$$

Choice2 $$\frac{S_2(l)\xrightarrow{(\alpha,v)}_c S^{'}{}_2(l')}{(S_1 + S_2)(l)\xrightarrow{(\alpha,v)}_c S^{'}{}_2(l')}$$

Constant $$\frac{S(l)\xrightarrow{(\alpha,S:[op(l,\kappa))]}_c S'(l')}{C(l)\xrightarrow{(\alpha,C:[op(l,\kappa))]}_c S'(l')}$$ with $C \overset{def}{=} S$

Semantics: cooperation rules

Semantics: cooperation rules

$$\text{coop1} \quad \frac{P_1 \xrightarrow{(\alpha,v)}_c P_1'}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,v)}_c P_1' \underset{\mathcal{L}}{\bowtie} P_2} \quad \text{with } \alpha \notin \mathcal{L}$$

Semantics: cooperation rules

$$\text{coop1} \quad \frac{P_1 \xrightarrow{(\alpha,v)}_c P_1'}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,v)}_c P_1' \underset{\mathcal{L}}{\bowtie} P_2} \quad \text{with } \alpha \notin \mathcal{L}$$

$$\text{coop2} \quad \frac{P_2 \xrightarrow{(\alpha,v)}_c P_2'}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,v)}_c P_1 \underset{\mathcal{L}}{\bowtie} P_2'} \quad \text{with } \alpha \notin \mathcal{L}$$

Semantics: cooperation rules

$$\texttt{coop1} \qquad \frac{P_1 \xrightarrow{(\alpha,v)}_c P_1'}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,v)}_c P_1' \underset{\mathcal{L}}{\bowtie} P_2} \qquad \text{with } \alpha \notin \mathcal{L}$$

$$\texttt{coop2} \qquad \frac{P_2 \xrightarrow{(\alpha,v)}_c P_2'}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,v)}_c P_1 \underset{\mathcal{L}}{\bowtie} P_2'} \qquad \text{with } \alpha \notin \mathcal{L}$$

$$\texttt{coopFinal} \qquad \frac{P_1 \xrightarrow{(\alpha,v_1)}_c P_1' \qquad P_2 \xrightarrow{(\alpha,v_2)}_c P_2'}{P_1 \underset{\mathcal{L}}{\bowtie} P_2 \xrightarrow{(\alpha,v_1::v_2)}_c P_1' \underset{\mathcal{L}}{\bowtie} P_2'} \qquad \text{with } \alpha \in \mathcal{L}$$

Semantics: rates and transition system

In order to derive the rates we consider the *stochastic relation* $\rightarrow_S \subseteq \mathcal{P} \times \Gamma \times \mathcal{P}$, with $\gamma \in \Gamma := (\alpha, r)$ and $r \in \mathbb{R}^+$.

Semantics: rates and transition system

In order to derive the rates we consider the *stochastic relation* $\rightarrow_S \subseteq \mathcal{P} \times \Gamma \times \mathcal{P}$, with $\gamma \in \Gamma := (\alpha, r)$ and $r \in \mathbb{R}^+$.

The relation is defined in terms of the previous one:

Semantics: rates and transition system

In order to derive the rates we consider the *stochastic relation* $\rightarrow_S \subseteq \mathcal{P} \times \Gamma \times \mathcal{P}$, with $\gamma \in \Gamma := (\alpha, r)$ and $r \in \mathbb{R}^+$.

The relation is defined in terms of the previous one:

$$\texttt{Final} \quad \frac{P \xrightarrow{(\alpha_j, v)}_c P'}{\langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, Comp, P\rangle \xrightarrow{(\alpha_j, r_{\alpha_j})}_S \langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, Comp, P'\rangle}$$

Semantics: rates and transition system

In order to derive the rates we consider the *stochastic relation* $\rightarrow_S \subseteq \mathcal{P} \times \Gamma \times \mathcal{P}$, with $\gamma \in \Gamma := (\alpha, r)$ and $r \in \mathbb{R}^+$.

The relation is defined in terms of the previous one:

$$\texttt{Final} \quad \frac{P \xrightarrow{(\alpha_j, v)}_c P'}{\langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, Comp, P \rangle \xrightarrow{(\alpha_j, r_{\alpha_j})}_S \langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, Comp, P' \rangle}$$

r_{α_j} represents the parameter of an exponential distribution and the dynamic behaviour is determined by a race condition.

Semantics: rates and transition system

In order to derive the rates we consider the *stochastic relation* $\rightarrow_S \subseteq \mathcal{P} \times \Gamma \times \mathcal{P}$, with $\gamma \in \Gamma := (\alpha, r)$ and $r \in \mathbb{R}^+$.

The relation is defined in terms of the previous one:

$$\texttt{Final} \quad \frac{P \xrightarrow{(\alpha_j, v)}_c P'}{\langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, Comp, P \rangle \xrightarrow{(\alpha_j, r_{\alpha_j})}_S \langle \mathcal{V}, \mathcal{N}, \mathcal{K}, \mathcal{F}_R, Comp, P' \rangle}$$

r_{α_j} represents the parameter of an exponential distribution and the dynamic behaviour is determined by a race condition.
The rate r_{α_j} is defined as $f_{\alpha_j}(v, \mathcal{N})/h$.

The abstraction

- Each species i is described by a Bio-PEPA component C_i.

The abstraction

- Each species i is described by a Bio-PEPA component C_i.
- Each reaction j is associated with an action type α_j and its dynamics is described by a specific function f_{α_j}.

The abstraction

- Each species *i* is described by a Bio-PEPA component C_i.
- Each reaction *j* is associated with an action type α_j and its dynamics is described by a specific function f_{α_j}.

  Given a reaction *j*, all the species/components cooperate together along the action type α_j and consequently, reactants decrease their levels, while products increase them. All the reactions are abstracted by cooperation.

The abstraction

- Each species i is described by a Bio-PEPA component C_i.
- Each reaction j is associated with an action type α_j and its dynamics is described by a specific function f_{α_j}.

  Given a reaction j, all the species/components cooperate together along the action type α_j and consequently, reactants decrease their levels, while products increase them. All the reactions are abstracted by cooperation.
- Compartments are static and represented by names indicating the location of species.

The abstraction

- Each species *i* is described by a Bio-PEPA component C_i.
- Each reaction *j* is associated with an action type α_j and its dynamics is described by a specific function f_{α_j}.

  Given a reaction *j*, all the species/components cooperate together along the action type α_j and consequently, reactants decrease their levels, while products increase them. All the reactions are abstracted by cooperation.
- Compartments are static and represented by names indicating the location of species.

The species components are then composed together to describe the behaviour of the system.

Example: Michaelis-Menten

The reaction $S \xrightarrow{E} P$ represents the enzymatic reaction from the substrate S to the product P with enzyme E.

Example: Michaelis-Menten

The reaction $S \xrightarrow{E} P$ represents the enzymatic reaction from the substrate S to the product P with enzyme E.

The dynamics is described by the law $\frac{v \times E \times S}{(K+S)}$.

Example: Michaelis-Menten

The reaction $S \xrightarrow{E} P$ represents the enzymatic reaction from the substrate S to the product P with enzyme E.

The dynamics is described by the law $\frac{v \times E \times S}{(K+S)}$.

$$\begin{aligned} S &\overset{def}{=} (\alpha, 1){\downarrow}S \\ E &\overset{def}{=} (\alpha, 1) \oplus E \\ P &\overset{def}{=} (\alpha, 1){\uparrow}P \end{aligned}$$

$$(S(l_{S0}) \underset{\{\alpha\}}{\bowtie} E(l_{E0})) \underset{\{\alpha\}}{\bowtie} P(l_{P0})$$

Example: Competitive Inhibition

Binding of the inhibitor to the enzyme prevents binding of the substrate and vice versa.

$$EI \longleftrightarrow S + E + I \longleftrightarrow SE \longrightarrow P + E$$

Example: Competitive Inhibition

Binding of the inhibitor to the enzyme prevents binding of the substrate and vice versa.

$$EI \longleftrightarrow S + E + I \longleftrightarrow SE \longrightarrow P + E$$

Under QSSA (the intermediate species *SE* and *EI* are constant) we can approximate the reactions above by a unique reaction

$$S \xrightarrow{E,I:f_I} P \qquad \text{with rate } f_I = \frac{w \times S \times E}{S + K_M(1 + \frac{I}{K_I})}$$

where w: turnover number (catalytic constant),
K_M: Michaelis constant and K_I: inhibition constant.

Example: Competitive Inhibition (2)

The specification in Bio-PEPA is:

$$S = (\alpha, 1)\downarrow S \qquad P = (\alpha, 1)\uparrow P \qquad E = (\alpha, 1) \oplus E \qquad I = (\alpha, 1) \ominus I$$

Example: Competitive Inhibition (2)

The specification in Bio-PEPA is:

$$S = (\alpha, 1)\!\downarrow\! S \qquad P = (\alpha, 1)\!\uparrow\! P \qquad E = (\alpha, 1) \oplus E \qquad I = (\alpha, 1) \ominus I$$

The system is described by

$$\left(\left(S(l_{S0}) \underset{\{\alpha\}}{\bowtie} E(l_{E0})\right) \underset{\{\alpha\}}{\bowtie} I(l_{I0})\right) \underset{\{\alpha\}}{\bowtie} P(l_{P0})$$

Example: Competitive Inhibition (2)

The specification in Bio-PEPA is:

$$S = (\alpha, 1){\downarrow}S \qquad P = (\alpha, 1){\uparrow}P \qquad E = (\alpha, 1) \oplus E \qquad I = (\alpha, 1) \ominus I$$

The system is described by

$$\left(\left(S(l_{S0}) \underset{\{\alpha\}}{\bowtie} E(l_{E0})\right) \underset{\{\alpha\}}{\bowtie} I(l_{I0})\right) \underset{\{\alpha\}}{\bowtie} P(l_{P0})$$

with functional rate

$$f_\alpha = \frac{w \times S \times E}{S + K_M(1 + \frac{I}{K_I})}$$

Equivalence relations

We are seeking to define a number of equivalence relations. for BioPEPA — both those that are expected from the computer science perspective and those that are useful from the biological perspective.

Equivalence relations

We are seeking to define a number of equivalence relations. for BioPEPA — both those that are expected from the computer science perspective and those that are useful from the biological perspective.

From the computer science perspective we have defined an isomorphism and a (strong) bisimulation.

Equivalence relations

We are seeking to define a number of equivalence relations. for BioPEPA — both those that are expected from the computer science perspective and those that are useful from the biological perspective.

From the computer science perspective we have defined an isomorphism and a (strong) bisimulation.

From the biological perspective. we are investigating the situations in which biologists regard models or elements of models to be equivalent, particularly when this is employed for model simplification.

Analysis

A Bio-PEPA system is a formal, intermediate and compositional representation of the system.

Analysis

A Bio-PEPA system is a formal, intermediate and compositional representation of the system.

From it we can obtain

Analysis

A Bio-PEPA system is a formal, intermediate and compositional representation of the system.

From it we can obtain

- a CTMC (with levels)

Analysis

A Bio-PEPA system is a formal, intermediate and compositional representation of the system.

From it we can obtain

- a CTMC (with levels)
- a ODE system for simulation and other kinds of analysis

Analysis

A Bio-PEPA system is a formal, intermediate and compositional representation of the system.

From it we can obtain

- a CTMC (with levels)
- a ODE system for simulation and other kinds of analysis
- a Gillespie model for stochastic simulation

Analysis

A Bio-PEPA system is a formal, intermediate and compositional representation of the system.

From it we can obtain

- a CTMC (with levels)
- a ODE system for simulation and other kinds of analysis
- a Gillespie model for stochastic simulation
- a PRISM model for model checking

Analysis

A Bio-PEPA system is a formal, intermediate and compositional representation of the system.

From it we can obtain

- a CTMC (with levels)
- a ODE system for simulation and other kinds of analysis
- a Gillespie model for stochastic simulation
- a PRISM model for model checking

Each of these kinds of analysis can be of help for studying different aspects of the biological model. Moreover we are exploring how they can be used in conjunction.

Outline

Process Calculi

SPA

SPA for Systems Biology

Bio-PEPA

Model definition

The syntax and semantics

Equivalences and Analysis

Examples

Genetic network with negative feedback loop

Goldbeter's model

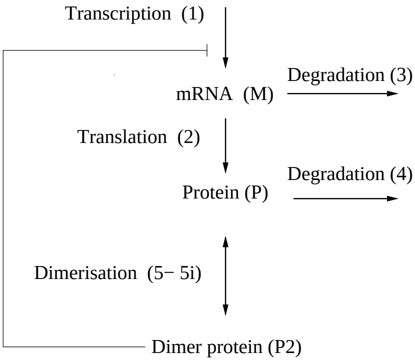
Conclusions

The biological model

Consider a genetic network with negative feedback through dimers.

The biological model

Consider a genetic network with negative feedback through dimers.



Species and reactions

The biological entities are:

- the *mRNA molecule* (*M*),
- the protein in monomer form (*P*) and
- the protein in dimeric form (*P2*).

Species and reactions

The biological entities are:

- the *mRNA molecule* (*M*),
- the protein in monomer form (*P*) and
- the protein in dimeric form (*P2*).

All the reactions are described by mass action kinetics with the exception of the first reaction, that has an inhibition kinetics.

Translation into Bio-PEPA

Definition of the list $\mathcal{N}$

$$[M : N_M, h_M; \quad P : N_P, h_P; \quad P2 : N_{P2}, h_{P2}]$$

Translation into Bio-PEPA

Definition of the list $\mathcal{N}$

$$[M : N_M, h_M; \quad P : N_P, h_P; \quad P2 : N_{P2}, h_{P2}]$$

Definition of functional rates

$$\begin{aligned}
f_{\alpha_1} &= \frac{v}{K_M + P2} \\
f_{\alpha_2} &= fMA(k_2) \qquad f_{\alpha_3} = fMA(k_3) \qquad f_{\alpha_4} = fMA(k_4) \\
f_{\alpha_5} &= fMA(k_5) \qquad f_{\alpha 5\_Inv} = fMA(k_{5\_Inv})
\end{aligned}$$

Translation into Bio-PEPA (cont.)

Definition of the system components

$$
\begin{array}{lcl}
\mathrm{M} & = & (\alpha_1,1) \uparrow \mathrm{M} + (\alpha_2,1) \oplus \mathrm{M} + (\alpha_3,1) \downarrow \mathrm{M};\\
\mathrm{P} & = & (\alpha_2,2) \uparrow \mathrm{P} + (\alpha_4,1) \downarrow \mathrm{P} + (\alpha_5,2) \downarrow \mathrm{P} + (\alpha_{5\_Inv},2) \uparrow \mathrm{P});\\
\mathrm{P2} & = & (\alpha_1,1) \ominus \mathrm{P2} + (\alpha_{5\_Inv},1) \downarrow \mathrm{P2} + (\alpha_5,1) \uparrow \mathrm{P2};\\
\mathrm{Res} & = & (\alpha_3,1) \odot \mathrm{Res} + (\alpha_4,1) \odot \mathrm{Res};\\
\mathrm{CF} & = & (\alpha_1,1) \odot \mathrm{CF};
\end{array}
$$

Translation into Bio-PEPA (cont.)

Definition of the system components

$$
\begin{array}{lcl}
\mathrm{M} & = & (\alpha_1,1)\uparrow \mathrm{M} + (\alpha_2,1)\oplus \mathrm{M} + (\alpha_3,1)\downarrow \mathrm{M};\\
\mathrm{P} & = & (\alpha_2,2)\uparrow \mathrm{P} + (\alpha_4,1)\downarrow \mathrm{P} + (\alpha_5,2)\downarrow \mathrm{P} + (\alpha_{5\_\mathit{Inv}},2)\uparrow \mathrm{P});\\
\mathrm{P2} & = & (\alpha_1,1)\ominus \mathrm{P2} + (\alpha_{5\_\mathit{Inv}},1)\downarrow \mathrm{P2} + (\alpha_5,1)\uparrow \mathrm{P2};\\
\mathrm{Res} & = & (\alpha_3,1)\odot \mathrm{Res} + (\alpha_4,1)\odot \mathrm{Res};\\
\mathrm{CF} & = & (\alpha_1,1)\odot \mathrm{CF};
\end{array}
$$

Definitions of the system

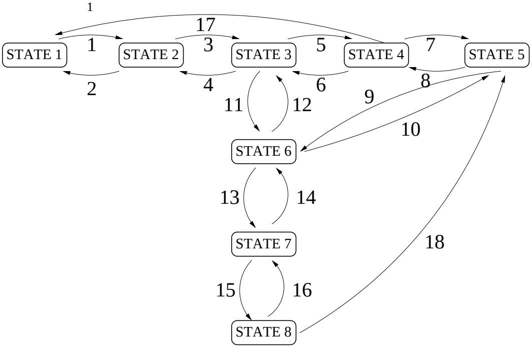
$$
((((CF(1) \underset{\{\alpha_1\}}{\bowtie} M(0)) \underset{\{\alpha_2\}}{\bowtie} P(0)) \underset{\{\alpha_5,\alpha_{5\_\mathit{Inv}}\}}{\bowtie} P2(0)) \underset{\{\alpha_3,\alpha_4\}}{\bowtie} Res(0)
$$

Analysis: the CTMC with levels

For **2 levels**, the CTMC consists of **8 states** and **18 transitions**.

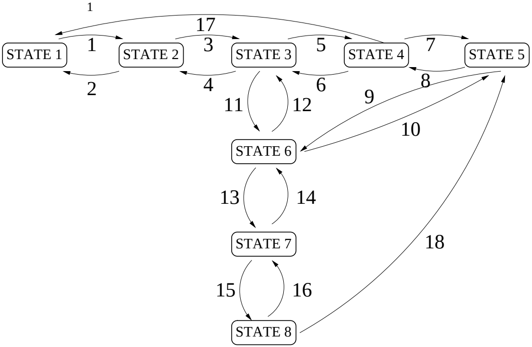
Analysis: the CTMC with levels

For **2 levels**, the CTMC consists of **8 states** and **18 transitions**.



Analysis: the CTMC with levels

For **2 levels**, the CTMC consists of **8 states** and **18 transitions**.



States are $(CF(l_1), M(l_2), P(l_3), P2(l_4), RES(l_5))$, with levels $l_1 \ldots l_5$.

Analysis: derivation of the ODE system

The stoichiometry matrix *D* associated with the system is

	α_1	α_2	α_3	α_4	α_5	α_{5_Inv}	
CF	0	0	0	0	0	0	x_{CF}
Res	0	0	0	0	0	0	x_{Res}
M	+1	0	-1	0	0	0	x_1
P	0	+1	0	-1	-2	+2	x_2
P2	0	0	0	0	+1	-1	x_3

Analysis: derivation of the ODE system

The stoichiometry matrix *D* associated with the system is

	α_1	α_2	α_3	α_4	α_5	α_{5_Inv}	
CF	0	0	0	0	0	0	x_{CF}
Res	0	0	0	0	0	0	x_{Res}
M	+1	0	-1	0	0	0	x_1
P	0	+1	0	-1	-2	+2	x_2
P2	0	0	0	0	+1	-1	x_3

The kinetic law vector is

$$w^T = (\frac{v \times x_{CF}}{K_M + x_3}; \quad k_2 \times x_1; k3 \times x_1; \quad k4 \times x_2; \quad k5 \times {x_2}^2; \quad k5\_Inv \times x_3)$$

Analysis: derivation of ODEs (cont.)

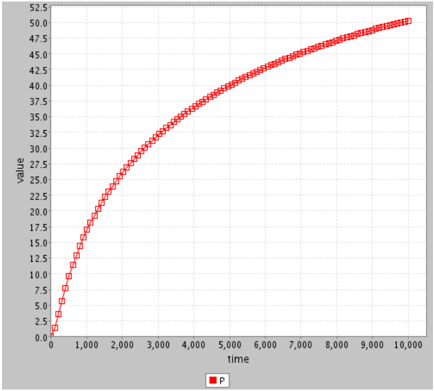
The system of ODEs is obtained as $\frac{d\bar{x}}{dt} = D \times w$:

$$\begin{aligned}
\frac{dx_1}{dt} &= \frac{v \times 1}{K_M + x_3} - k3 \times x_1 \\
\frac{dx_2}{dt} &= k2 \times x_1 - k4 \times x_2 - 2 \times k5 \times {x_2}^2 + 2 \times k5\_Inv \times x_3 \\
\frac{dx_2}{dt} &= k5 \times {x_2}^2 - k5\_Inv \times x_3
\end{aligned}$$

Analysis: stochastic simulation

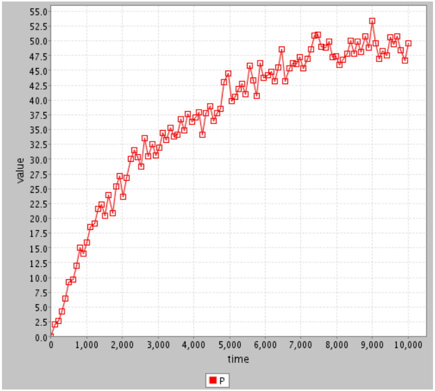
The derivation of the Gillespie model is made by creating molecules corresponding to each species and defining the possible reactions with appropriate adjustment of kinetic rates.

Simulation results



ODE results

Simulation results



Stochastic simulation results (10 runs)

PRISM model

Each species is represented as a PRISM module.

PRISM model

Each species is represented as a PRISM module.
For example, the protein is represented as:

module p
p: [0..Np] init 0;
$[Translation]\ p < Np \rightarrow (p' = p + 1);$
$[DegradationP]\ p > 0 \rightarrow (p' = p - 1);$
$[Dimerization]\ p > 1 \rightarrow (p' = p - 2);$
$[DimerizationInv]\ p < (Np - 1) \rightarrow (p' = p + 2);$
endmodule

PRISM model (cont.)

An additional module is needed to capture the kinetic rates.

module Functional_rates
dummy: bool **init** true;
$[\mathit{Transcription}]\ m < Nm \rightarrow (v/(K + p2 * h_{p2}) * h_{p2}) : (\mathit{dummy}' = \mathit{dummy});$
$[\mathit{Translation}]\ m > 0 \rightarrow (k2 * m * h_m/h_m) : (\mathit{dummy}' = \mathit{dummy});$
$[\mathit{DegradationmRNA}]\ m > 0 \rightarrow (k3 * m * h_m/h_m) : (\mathit{dummy}' = \mathit{dummy});$
$[\mathit{DegradationP}]\ p > 0 \rightarrow (k4 * p * h_p/h_p) : (\mathit{dummy}' = \mathit{dummy});$
$[\mathit{Dimerization}]\ p > 1 \rightarrow (k5 * p * h_p * p * h_p/h_p)(\mathit{dummy}' = \mathit{dummy});$
$[\mathit{DimerizationInv}]\ p2 > 0 \rightarrow (k5\_Inv * p2 * h_{p2}/h_{p2}) : (\mathit{dummy}' = \mathit{dummy});$
endmodule

PRISM analysis

PRISM analysis

- **Proportion of monomer P in total P (in terms of levels).**
  We need to define a reward structure in the PRISM file as:

  rewards
  $true : \frac{p}{(p+p2)};$
  endrewards

PRISM analysis

- **Proportion of monomer P in total P (in terms of levels).**
  We need to define a reward structure in the PRISM file as:

$$\textbf{rewards}$$
$$true : \frac{p}{(p+p2)};$$
$$\textbf{endrewards}$$

We can ask for the proportion of monomer P by using the query:

$$R = ?[I = T]$$

PRISM analysis

- **Proportion of monomer P in total P (in terms of levels).**

  We need to define a reward structure in the PRISM file as:

  rewards
  $true : \frac{p}{(p+p2)};$
  endrewards

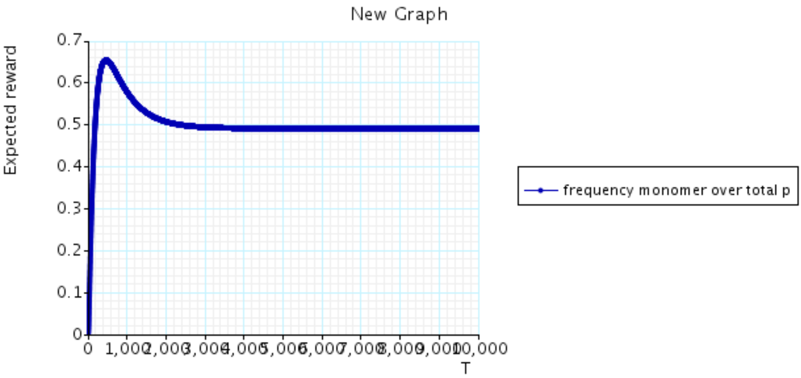
  We can ask for the proportion of monomer P by using the query:

  $$R = ?[I = T]$$

- **Probability that P is at level i at time T**

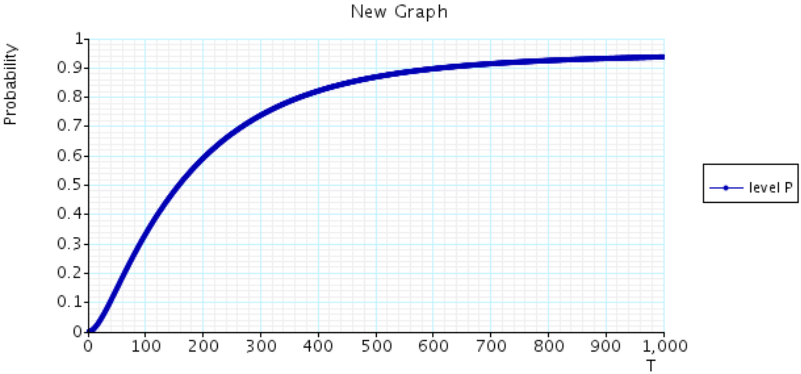
  $$P = ?[trueU[T, T]p = i]$$

PRISM results



monomer frequency

PRISM results



Probability monomer protein is at high level over time

Goldbeter's model [Goldbeter 91]

- Goldbeter's model describes the activity of the cyclin in the cell cycle.

Goldbeter's model [Goldbeter 91]

- Goldbeter's model describes the activity of the cyclin in the cell cycle.
- The cyclin promotes the activation of a cdk (cdc2) which in turn activates a cyclin protease.

Goldbeter's model [Goldbeter 91]

- Goldbeter's model describes the activity of the cyclin in the cell cycle.
- The cyclin promotes the activation of a cdk (cdc2) which in turn activates a cyclin protease.
- This protease promotes cyclin degradation.

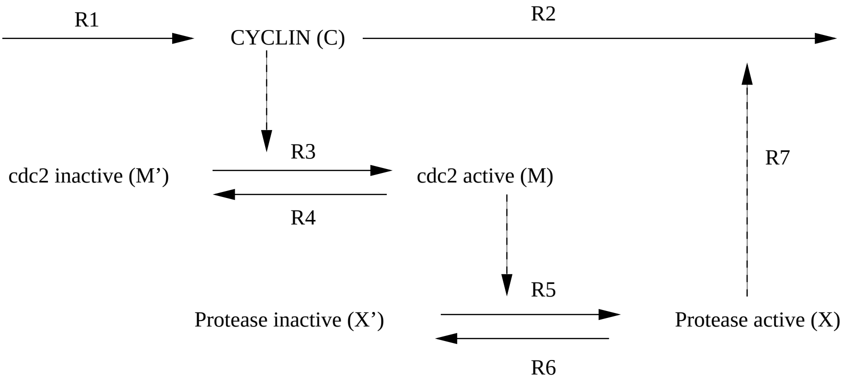
Goldbeter's model [Goldbeter 91]

- Goldbeter's model describes the activity of the cyclin in the cell cycle.
- The cyclin promotes the activation of a cdk (cdc2) which in turn activates a cyclin protease.
- This protease promotes cyclin degradation.
- This leads to a negative feedback loop.

Goldbeter's model [Goldbeter 91]

- Goldbeter's model describes the activity of the cyclin in the cell cycle.
- The cyclin promotes the activation of a cdk (cdc2) which in turn activates a cyclin protease.
- This protease promotes cyclin degradation.
- This leads to a negative feedback loop.
- In the model most of the kinetic laws are of kind Michaelis-Menten and this can be reflected in the Bio-PEPA model.

The biological model



The biological model (2)

There are three different biological species involved:

The biological model (2)

There are three different biological species involved:

- cyclin, the protein protagonist of the cycle, C;

The biological model (2)

There are three different biological species involved:

- cyclin, the protein protagonist of the cycle, C;
- cdc2 kinase, in both active (i.e. dephosphorylated) and inactive form (i.e. phosphorylated). The variables used to represent them are M and M', respectively;

The biological model (2)

There are three different biological species involved:

- cyclin, the protein protagonist of the cycle, C;
- cdc2 kinase, in both active (i.e. dephosphorylated) and inactive form (i.e. phosphorylated). The variables used to represent them are M and M', respectively;
- cyclin protease, in both active (i.e. phosphorylated) and inactive form (i.e. dephosphorylated). The variable are X and X'.

Reactions

id	desc.	react.	prod.	mod.	kinetic laws
R1	creation of cyclin	-	C	-	vi
R2	degradation of cyclin	C	-	-	$kd \times C$
R3	activation of cdc2 kinase	M'	M	-	$\frac{C*V_{M1}}{(K_c+C)} \frac{M'}{(K_1+M')}$
R4	deactivation of cdc2 kinase	M	M'	-	$\frac{M\times V_2}{(K_2+M)}$
R5	activation of cyclin protease	X'	X	M	$\frac{X'\times M\times V_{M3}}{(K_3+X')}$
R6	deactivation of cyclin protease	X	X'	-	$\frac{X\times V_4}{K_4+X}$
R7	X triggered degradation of cyclin	C	-	X	$\frac{C\times v_d\times X}{C+K_d}$

R1 and R2 have Mass-Action kinetics, whereas all others are Michaelis-Menten.

Translation into Bio-PEPA

Definition of the set $\mathcal{N}$:

$$\mathcal{N} = [Res : 1, 1;\ CF : 1, 1;\ C : h_C, N_c;\ M : h_M, N_M; \\ M' : h_{M'}, N_{M'};\ X : h_X, N_X, ;\ X' : h_{X'}, N_{X'}]$$

Res and *CF* represent degradation and synthesis respectively.

Translation into Bio-PEPA

Definition of the set $\mathcal{N}$:

$$\mathcal{N} = [Res : 1, 1;\ CF : 1, 1;\ C : h_C, N_c;\ M : h_M, N_M;\\ M' : h_{M'}, N_{M'};\ X : h_X, N_X, ;\ X' : h_{X'}, N_{X'}]$$

Res and *CF* represent degradation and synthesis respectively.
Definition of functional rates ($\mathcal{F}$):

$$\begin{array}{lllllll} f_{\alpha_1} & = & fMA(v_i); & & f_{\alpha_2} & = & fMA(k_d); \\ f_{\alpha_4} & = & fMM(V_2, K_2); & & f_{\alpha_5} & = & fMM(V_3, K_3); \\ f_{\alpha_6} & = & fMM(V_4, K_4); & & f_{\alpha_7} & = & fMM(V_d, K_d); \end{array}$$

$$f_{\alpha_3} = \frac{v_1 \times C}{K_c + C} \frac{M'}{K1 + M'}$$

The Bio-PEPA system (2)

Definition of species components (*Comp*):

$$\begin{aligned}
C &= (\alpha_1, 1)\uparrow C + (\alpha_2, 1)\downarrow C + (\alpha_3, 1) \oplus C + (\alpha_7, 1)\downarrow C;\\
M' &= (\alpha_3, 1)\downarrow M' + (\alpha_4, 1)\uparrow M';\\
M &= (\alpha_3, 1)\uparrow M + (\alpha_4, 1)\downarrow M + (\alpha_5, 1) \oplus M;\\
X' &= (\alpha_5, 1)\downarrow X' + (\alpha_6, 1)\uparrow X';\\
X &= (\alpha_5, 1)\uparrow X + (\alpha_6, 1)\downarrow X + (\alpha_7, 1) \oplus X;\\
Res &= (\alpha_2, 1) \odot Res; \quad CF = (\alpha_1, 1) \odot CF;
\end{aligned}$$

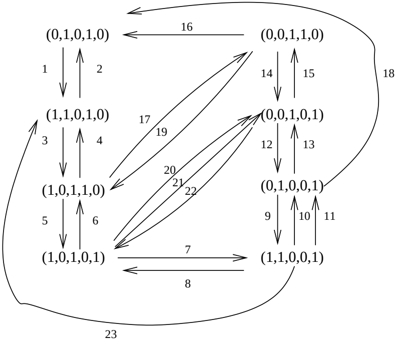
Definition of the model component (*P*):

$$C(l_{0C}) \underset{\{\alpha_3\}}{\bowtie} M(l_{0M}) \underset{\{\alpha_3,\alpha_4\}}{\bowtie} M^{'}(l_{0M'}) \underset{\{\alpha_5,\alpha_7\}}{\bowtie} X(l_{0X}) \underset{\{\alpha_5,\alpha_6\}}{\bowtie} X^{'}(l_{0X'})$$

$$\underset{\{\alpha_2\}}{\bowtie} Deg(0) \underset{\{\alpha_1\}}{\bowtie} CF(1)$$

Analysis: CTMC with 2 levels

Assume two levels for each species and initially C, M and X present (level 1) and the other elements not present (level 0).
The initial state is $(l_C(1), l_{M'}(0), l_M(1), l_{X'}(0), l_X(1))$.



Analysis: ODEs

The stoichiometry matrix D:

	R1	R2	R3	R4	R5	R6	R7	
C	+1	0	0	0	0	0	-1	x_C
M'	0	0	-1	+1	0	0	0	$x_{M'}$
M	0	0	+1	-1	0	0	0	x_M
X'	0	0	0	0	-1	+1	0	$x_{X'}$
X	0	0	0	0	+1	-1	0	x_X

Analysis: ODEs

The stoichiometry matrix D:

$$
\begin{array}{c|c|c|c|c|c|c|c|c}
 & \mathrm{R1} & \mathrm{R2} & \mathrm{R3} & \mathrm{R4} & \mathrm{R5} & \mathrm{R6} & \mathrm{R7} & \\
\hline
\mathrm{C} & +1 & 0 & 0 & 0 & 0 & 0 & -1 & x_C \\
M' & 0 & 0 & -1 & +1 & 0 & 0 & 0 & x_{M'} \\
\mathrm{M} & 0 & 0 & +1 & -1 & 0 & 0 & 0 & x_M \\
X' & 0 & 0 & 0 & 0 & -1 & +1 & 0 & x_{X'} \\
\mathrm{X} & 0 & 0 & 0 & 0 & +1 & -1 & 0 & x_X
\end{array}
$$

The vector that contains the kinetic laws is:

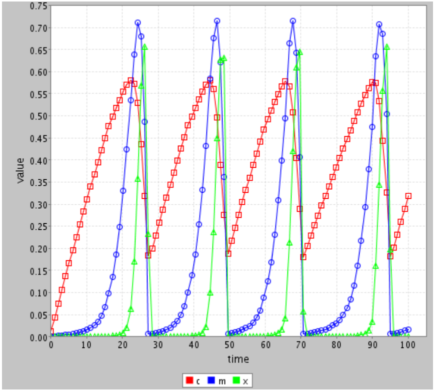
$$
\begin{aligned}
w = \Big(v_i \times 1, k_d \times x_C, \frac{V_{M1} \times x_C}{K_c + x_C} \frac{x_{M'}}{(K_1 + x_{M'})}, \frac{V_2 \times x_M}{(K_2 + x_M)}, \\
\frac{V_{M3} \times x_M \times x_{X'}}{(K_3 + x_{X'})}, \frac{V_4 \times x_X}{(K_4 + x_X)}, \frac{v_d \times x_C \times x_X}{(K_d + x_C)} \Big)
\end{aligned}
$$

Analysis: ODEs (2)

The system of ODEs is obtained as $\frac{d\bar{x}}{dt} = D \times w$, where $\bar{x}^T =: (x_C, x_{M'}, x_M, x_{X'}, x_X)$ is the vector of the species variables:

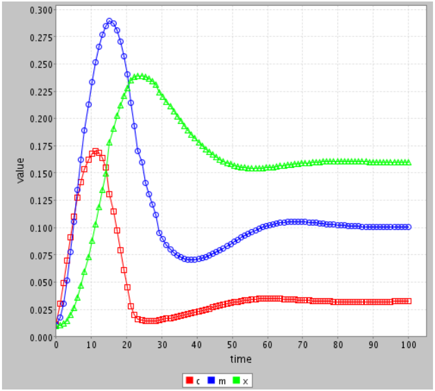
$$\begin{aligned}
\frac{dx_C}{dt} &= v_i \times 1 - k_d \times x_C - \frac{v_d \times x_C \times x_X}{(K_d + x_C)} \\
\frac{dx_{M'}}{dt} &= -\frac{V_{M1} \times x_C}{K_c + x_C} \frac{x_{M'}}{(K_1 + x_{M'})} + \frac{V_2 \times x_M}{(K_2 + x_M)} \\
\frac{dx_M}{dt} &= +\frac{V_{M1} \times x_C}{K_c + x_C} \frac{x_{M'}}{(K_1 + x_{M'})} - \frac{V_2 \times x_M}{(K_2 + x_M)} \\
\frac{dx_{X'}}{dt} &= -\frac{V_{M3} \times x_M \times x_{X'}}{(K_3 + x_{X'})} + \frac{V_4 \times x_X}{(K_4 + x_X)} \\
\frac{dx_X}{dt} &= \frac{V_{M3} \times x_M \times x_{X'}}{(K_3 + x_{X'})} - \frac{V_4 \times x_X}{(K_4 + x_X)}
\end{aligned}$$

ODE results



$$K_1 = K_2 = K_3 = K_4 = 0.02\mu M$$

ODE results



$$K_1 = K_2 = K_3 = K_4 = 40\mu M$$

Extension of the Goldbeter's model

- Gardner et al. [Gardner 98] proposed an extension of the Goldbeter's model in order to represent a control mechanism for the cell division cycle.

Extension of the Goldbeter's model

- Gardner et al. [Gardner 98] proposed an extension of the Goldbeter's model in order to represent a control mechanism for the cell division cycle.
- They introduce a protein that binds to and inhibits one of the proteins involved in the cell division cycle.

Extension of the Goldbeter's model

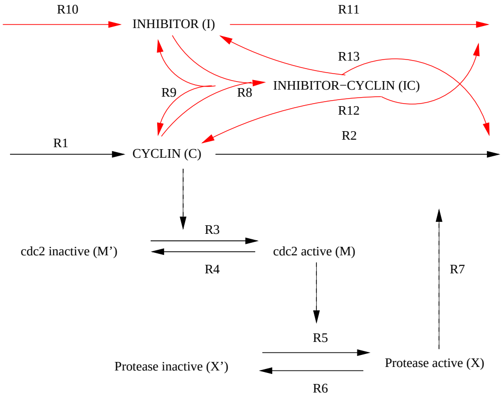
- Gardner et al. [Gardner 98] proposed an extension of the Goldbeter's model in order to represent a control mechanism for the cell division cycle.
- They introduce a protein that binds to and inhibits one of the proteins involved in the cell division cycle.
- This influences the start and the stop of the cell division and modulates the frequency of oscillations.

Extension of the Goldbeter's model

- Gardner et al. [Gardner 98] proposed an extension of the Goldbeter's model in order to represent a control mechanism for the cell division cycle.
- They introduce a protein that binds to and inhibits one of the proteins involved in the cell division cycle.
- This influences the start and the stop of the cell division and modulates the frequency of oscillations.

Several possible extensions were presented; we consider one of them.

Schema of the extended model



Extended Bio-PEPA system

$$
\begin{array}{lcl}
C & = & \cdots + (\alpha_8, 1)\downarrow C + (\alpha_9, 1)\uparrow C + (\alpha_{12}, 1)\uparrow C; \\
\vdots & & \vdots \\
Res & = & \cdots + (\alpha_{11}, 1) \odot Res; \quad CF = \cdots + (\alpha_{10}, 1) \odot CF; \\
I & = & (\alpha_8, 1)\downarrow I + (\alpha_9, 1)\uparrow I + (\alpha_{10}, 1)\uparrow I + (\alpha_{11}, 1)\downarrow I + (\alpha_{13}, 1)\uparrow I; \\
IC & = & (\alpha_8, 1)\uparrow IC + (\alpha_9, 1)\downarrow IC + (\alpha_{12}, 1)\downarrow IC + (\alpha_{13}, 1)\downarrow IC;
\end{array}
$$

New functional rates

$$
\begin{aligned}
f_{\alpha_8} &= v_s; \\
f_{\alpha_9} &= fMA(d_1); \\
f_{\alpha_{10}} &= fMA(a_1); \\
f_{\alpha_{11}} &= fMA(a_2); \\
f_{\alpha_{12}} &= fMA(\theta \times d_1); \\
f_{\alpha_{13}} &= fMA(\theta \times k_d)
\end{aligned}
$$

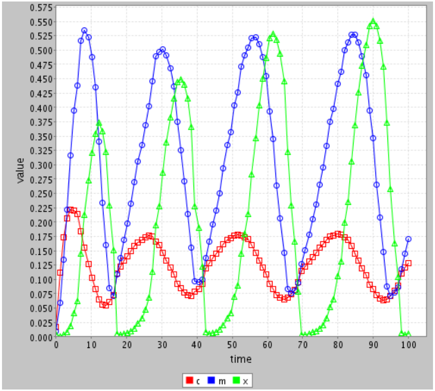
Complete Bio-PEPA system

$$C(l_{0C}) \underset{\{\alpha_3\}}{\bowtie} M(l_{0M}) \underset{\{\alpha_3,\alpha_4\}}{\bowtie} M^{'}(l_{0M'}) \underset{\{\alpha_5,\alpha_7\}}{\bowtie} X(l_{0X}) \underset{\{\alpha_5,\alpha_6\}}{\bowtie} X^{'}(l_{0X'}) \underset{\{\alpha_2\}}{\bowtie}$$

$$Deg(0) \underset{\{\alpha_1\}}{\bowtie} CF(1)$$

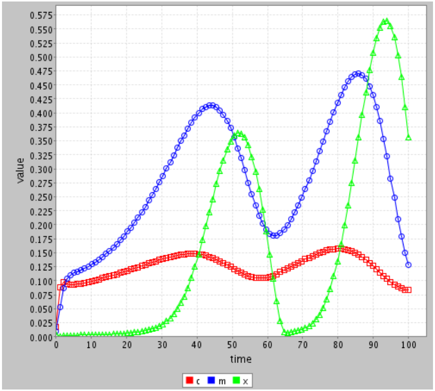
$$\underset{\{\alpha_8,\alpha_9,\alpha_{10},\alpha_{11}\}}{\bowtie} I(l_{0I}) \underset{\{\alpha_8,\alpha_9,\alpha_{12},\alpha_{13}\}}{\bowtie} IC(l_{0IC})$$

New ODE results



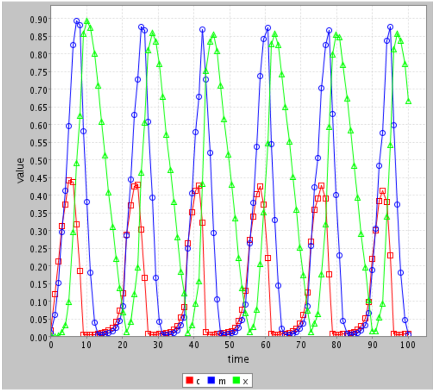
$a_1 = a_2 = 0.3$ and $v_s = 0.6$

New ODE results



$a_1 = a_2 = 0.7$ and $v_s = 1.4$

New ODE results



$$a_1 = a_2 = 0.05 \text{ and } v_s = 0.1$$

Outline

Process Calculi

SPA

SPA for Systems Biology

Bio-PEPA

Model definition

The syntax and semantics

Equivalences and Analysis

Examples

Genetic network with negative feedback loop

Goldbeter’s model

Conclusions

Conclusion: SPA for Systems Biology

Whilst the notation can be a challenge, the compositionality and precise interpretation of process algebras make them attractive for modelling biological signalling pathways.

Conclusion: SPA for Systems Biology

Whilst the notation can be a challenge, the compositionality and precise interpretation of process algebras make them attractive for modelling biological signalling pathways.

Choices in the design of the SPA such as the form of synchronisation which is incorporated has a strong influence on the way in which systems can be modelled.

Conclusion: SPA for Systems Biology

Whilst the notation can be a challenge, the compositionality and precise interpretation of process algebras make them attractive for modelling biological signalling pathways.

Choices in the design of the SPA such as the form of synchronisation which is incorporated has a strong influence on the way in which systems can be modelled.

The inclusion of stochastic information about the duration of actions/reactions creates a very natural mapping from SPA models to stochastic simulations at the molecular models.

Conclusion: SPA for Systems Biology

Whilst the notation can be a challenge, the compositionality and precise interpretation of process algebras make them attractive for modelling biological signalling pathways.

Choices in the design of the SPA such as the form of synchronisation which is incorporated has a strong influence on the way in which systems can be modelled.

The inclusion of stochastic information about the duration of actions/reactions creates a very natural mapping from SPA models to stochastic simulations at the molecular models.

However, such molecular mappings typically generate state spaces which are too large for other SPA analysis techniques.

Conclusions: Bio-PEPA

Bio-PEPA is a modification of the process algebra PEPA for the *modelling* and *analysis* of biochemical networks.

Conclusions: Bio-PEPA

Bio-PEPA is a modification of the process algebra PEPA for the *modelling* and *analysis* of biochemical networks.

Bio-PEPA allows us to represent explicitly features of biological networks, such as stoichiometry and general kinetic laws.

Conclusions: Bio-PEPA

Bio-PEPA is a modification of the process algebra PEPA for the *modelling* and *analysis* of biochemical networks.

Bio-PEPA allows us to represent explicitly features of biological networks, such as stoichiometry and general kinetic laws.

Moreover the reagent-centric, abstract style of modelling supports an integrative approach in which several different approaches to analysis may be applied to the same model.

Conclusions: Abstract Modelling

Abstract modelling offers a compromise between the individual-based and population-based views of systems which biologists commonly take.

Conclusions: Abstract Modelling

Abstract modelling offers a compromise between the individual-based and population-based views of systems which biologists commonly take.

Moveover we can undertake additional analysis based on the discretised population view.

Conclusions: Abstract Modelling

Abstract modelling offers a compromise between the individual-based and population-based views of systems which biologists commonly take.

Moveover we can undertake additional analysis based on the discretised population view.

The abstract Markovian models allow quantities of interest such as “response times” to be expressed as probability distributions rather than single estimates. This may allow better reflection of wet lab data which also shows variability.

Future directions

There are number of areas for on-going and future work. For example:

- The definition of bisimulations and equivalences.
- The extent to which the process algebra compositional structure can be exploited during model analysis, particularly in conjunction with model checking techniques.
- The issue of coping with unknown and uncertain values in experimental data.
- ...*and many more...*

Acknowledgements

The PEPA project has been funded by SERC, EPSRC and the CEC. Work on Bio-PEPA has been funded by BBSRC and EPSRC. In particular Jane Hillston and Federica Ciocchetta are supported by the CODA project, and the Centre for Systems Biology at Edinburgh.

We would like to thank our collaborators, Jeremy Bradley, Muffy Calder, Andrea Degasperi, Vashti Galpin, Stephen Gilmore, Nil Geisweiller, Maria Luisa Guerriero and Marco Stenico.

Thank you