Process algebras for quantitative analysis: Lecture 2 — Stochastic process algebras

Jane Hillston

School of Informatics
The University of Edinburgh
Scotland

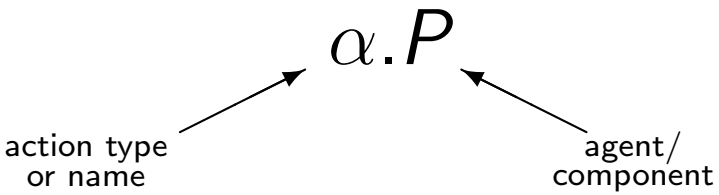
9th February 2010

Outline

Outline

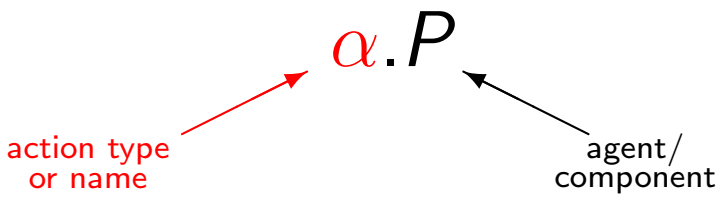
Process Algebra

- Models consist of agents which engage in actions.



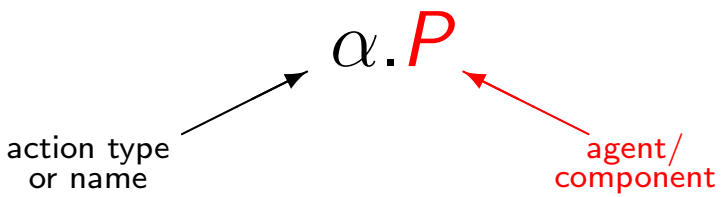
Process Algebra

- Models consist of agents which engage in actions.



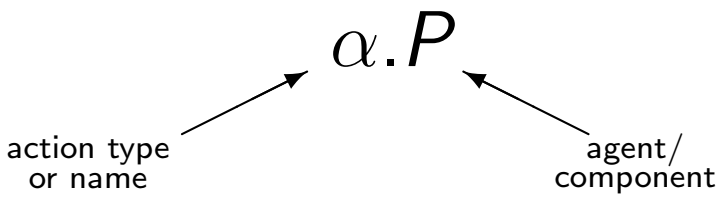
Process Algebra

- Models consist of agents which engage in actions.



Process Algebra

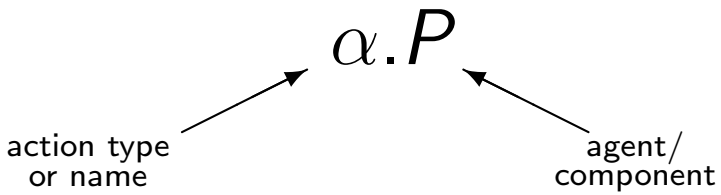
- Models consist of agents which engage in actions.



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

Process Algebra

- Models consist of agents which engage in actions.

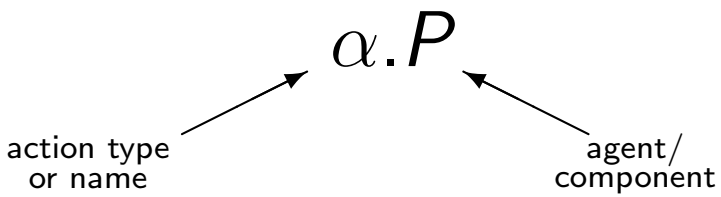


- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

Process algebra model

Process Algebra

- Models consist of agents which engage in actions.

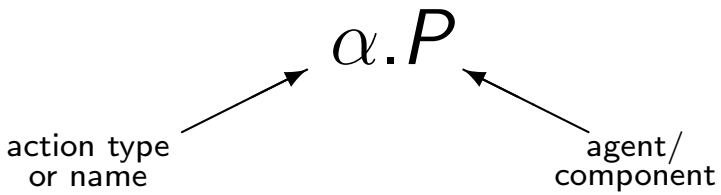


- The structured operational (interleaving) semantics of the language is used to generate a labelled transition system.

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

Process Algebra

- Models consist of agents which engage in actions.



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

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

Example

Consider a web server which offers html pages for download:

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

Example

Consider a web server which offers html pages for download:

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

Its clients might be 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.

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

Example

Consider a web server which offers html pages for download:

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

Its clients might be 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.

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

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

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

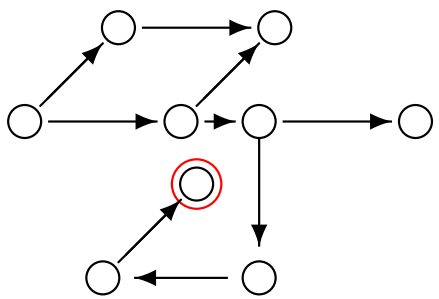
Qualitative Analysis

- 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.

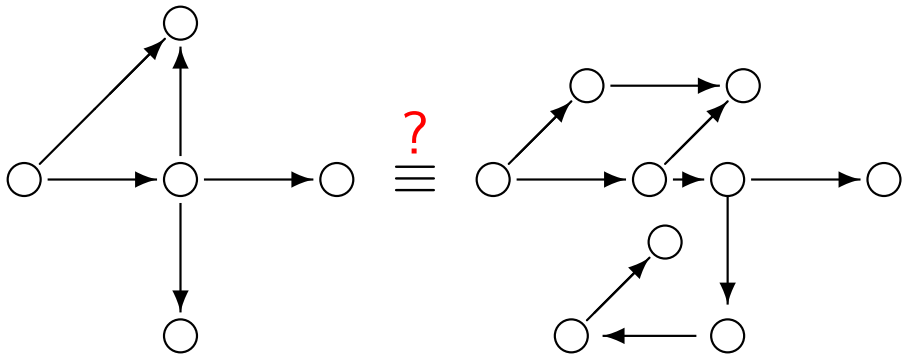
Will the system arrive in a particular state?



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.

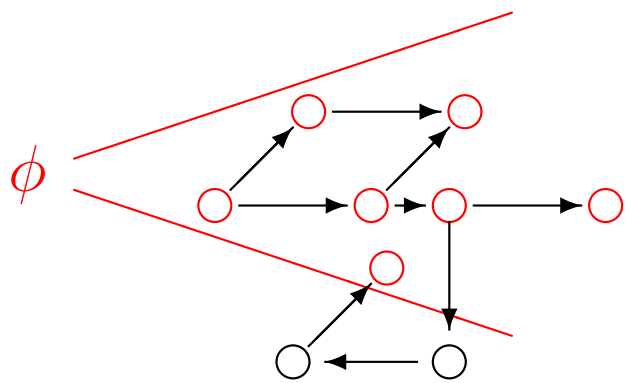
Does system behaviour match its specification?



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?



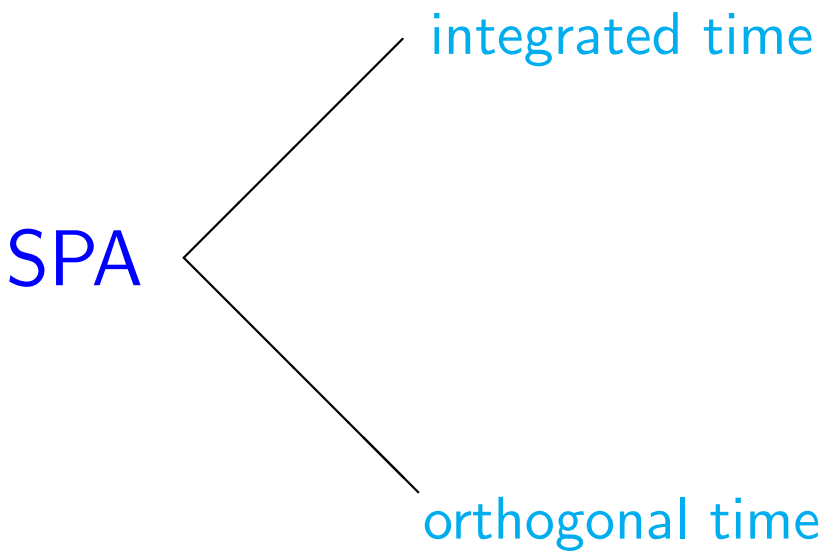
Stochastic process algebras

Process algebras where models are decorated with quantitative information used to generate a stochastic process are stochastic process algebras (SPA).

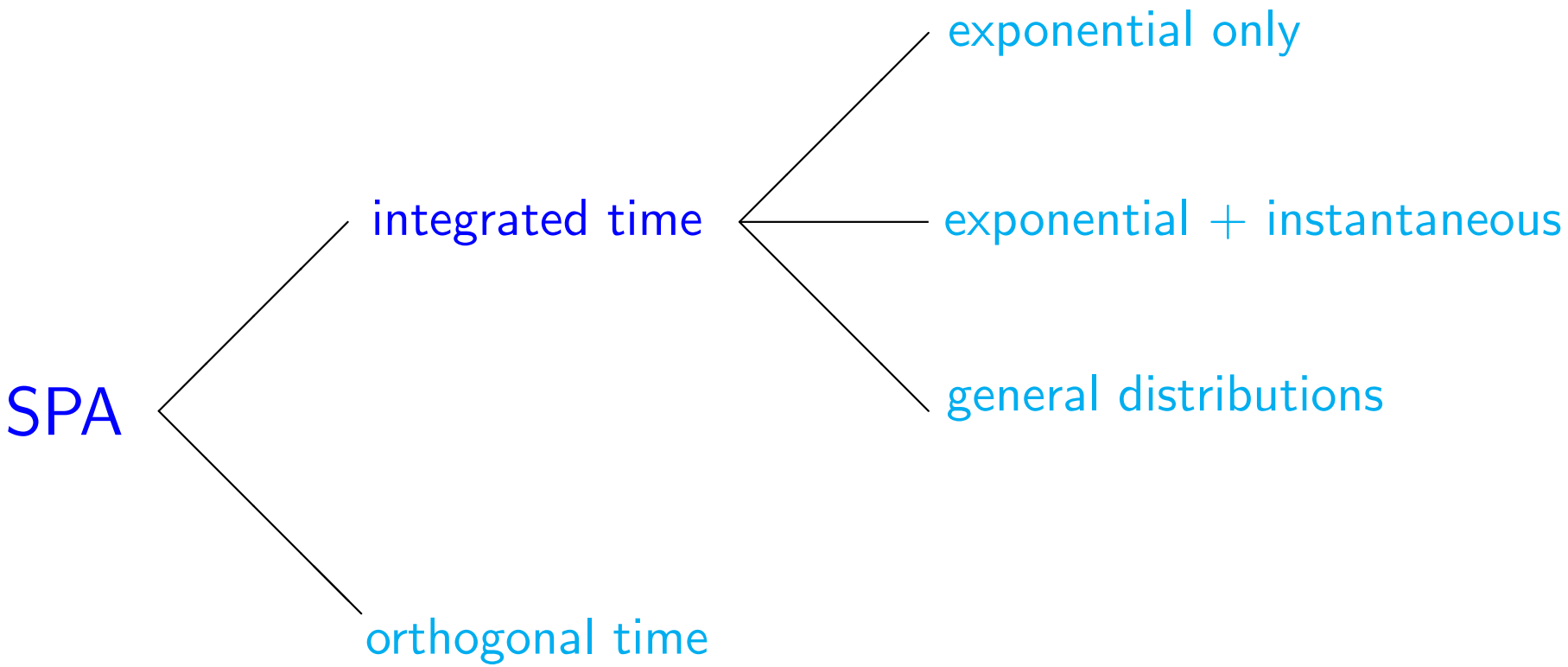
SPA Languages

SPA

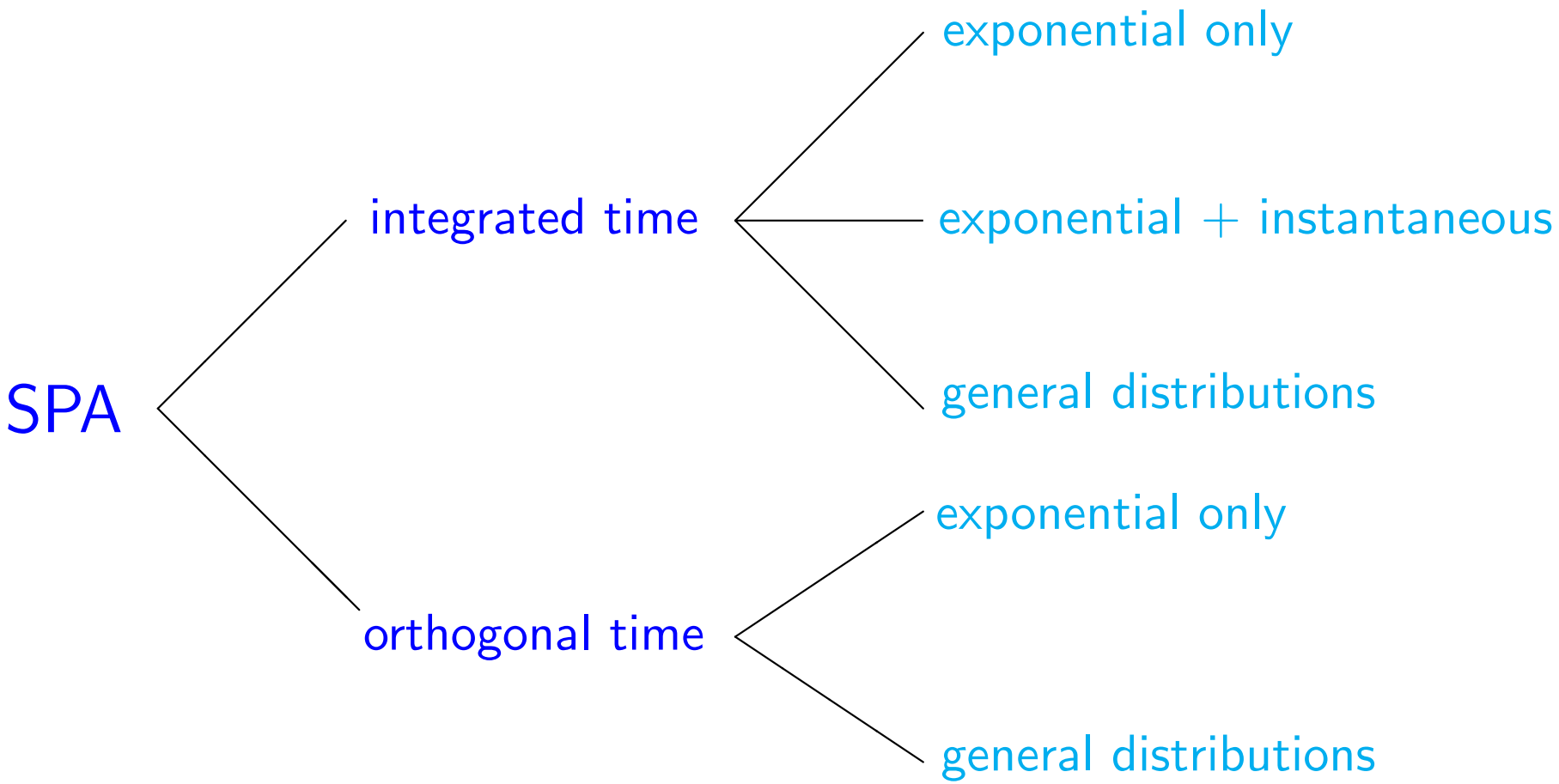
SPA Languages



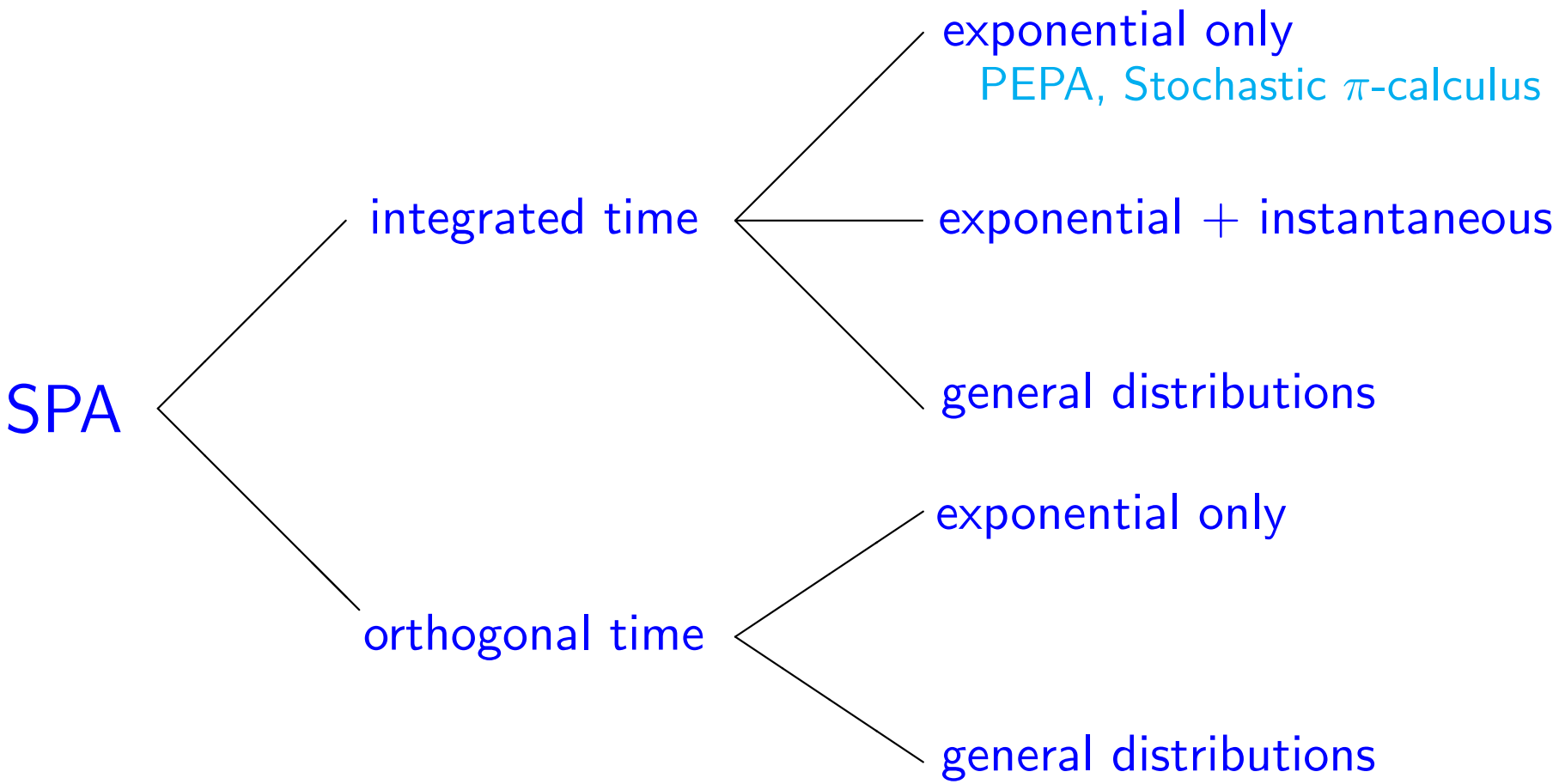
SPA Languages



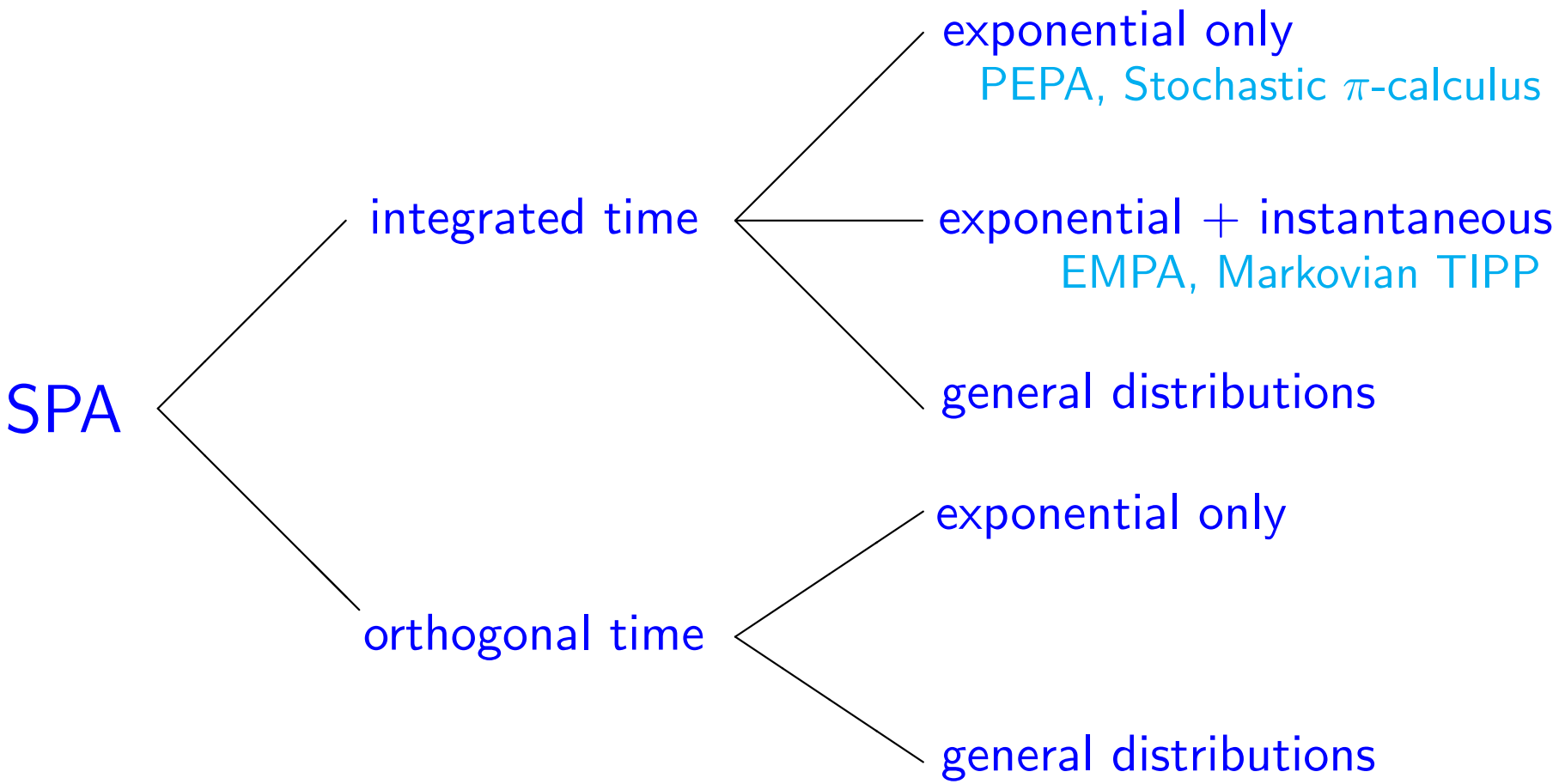
SPA Languages



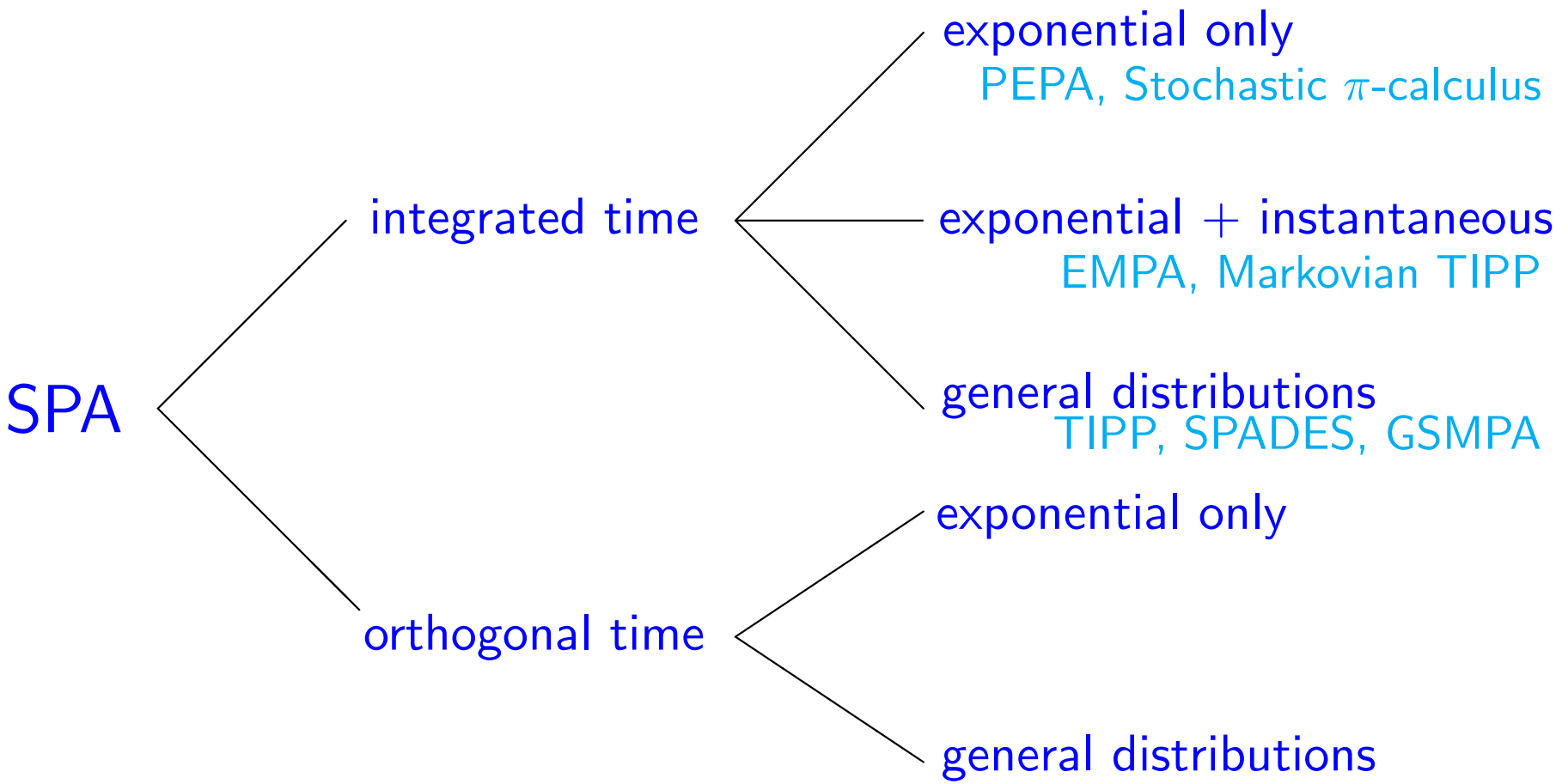
SPA Languages



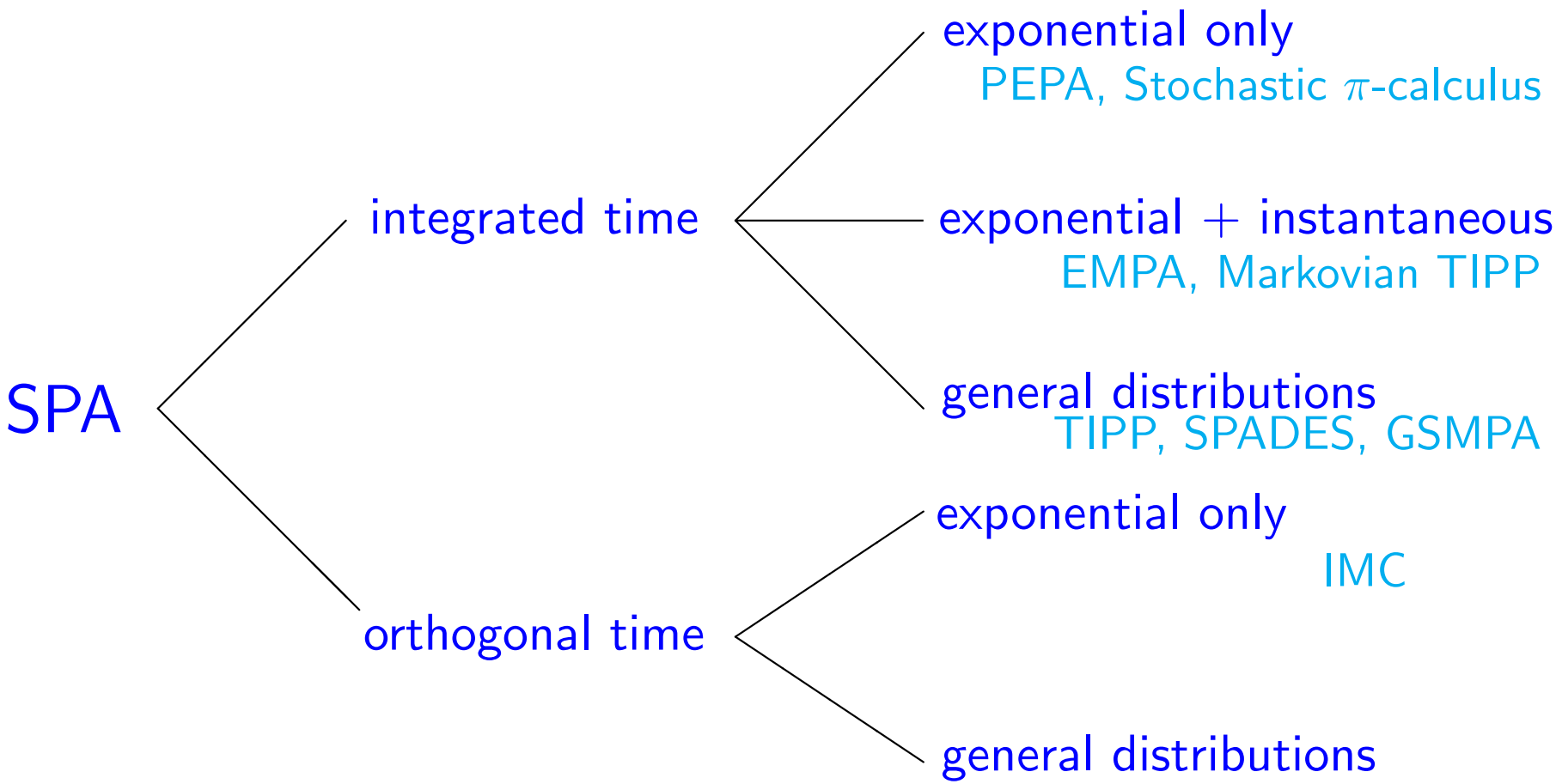
SPA Languages



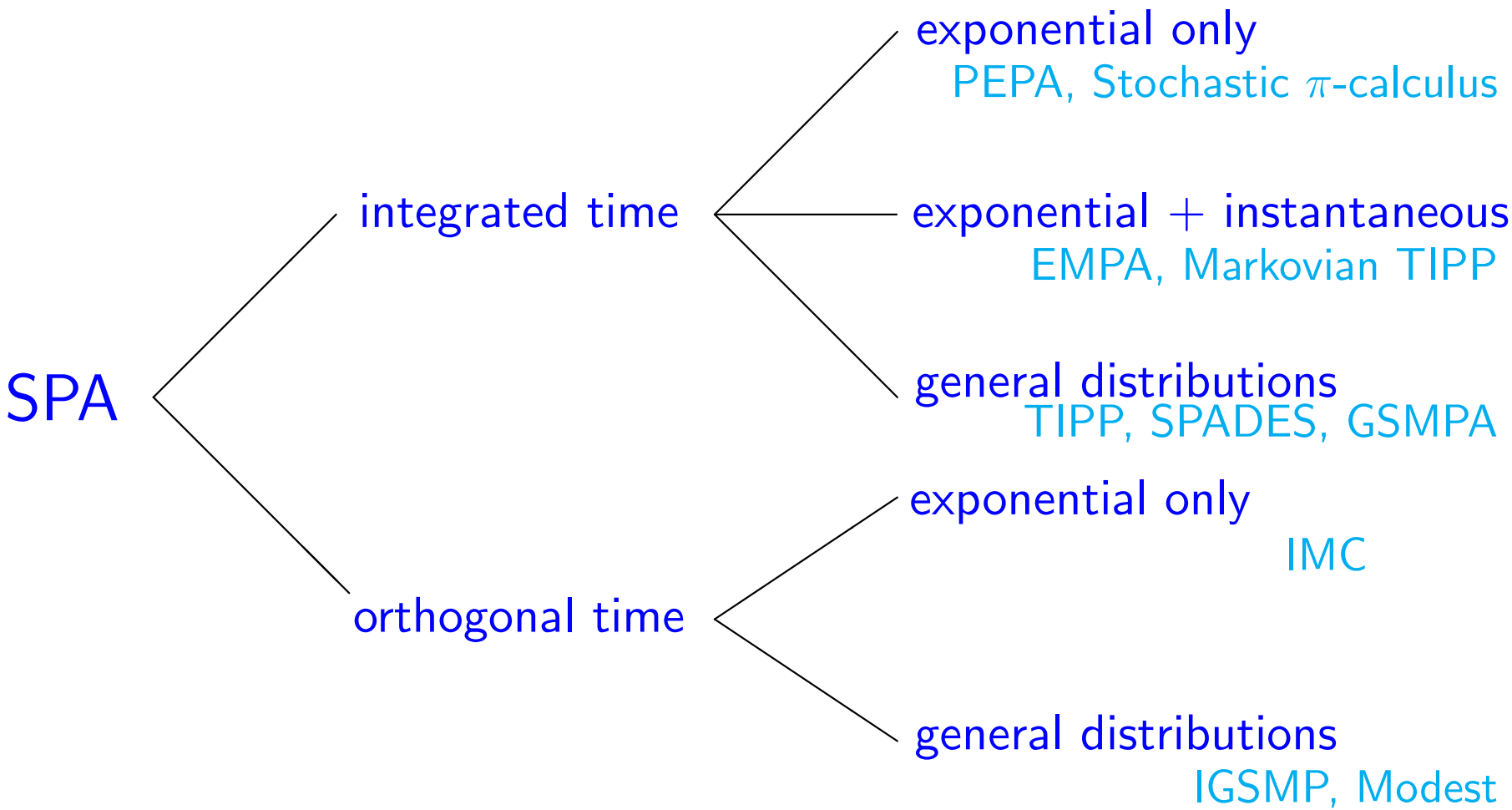
SPA Languages



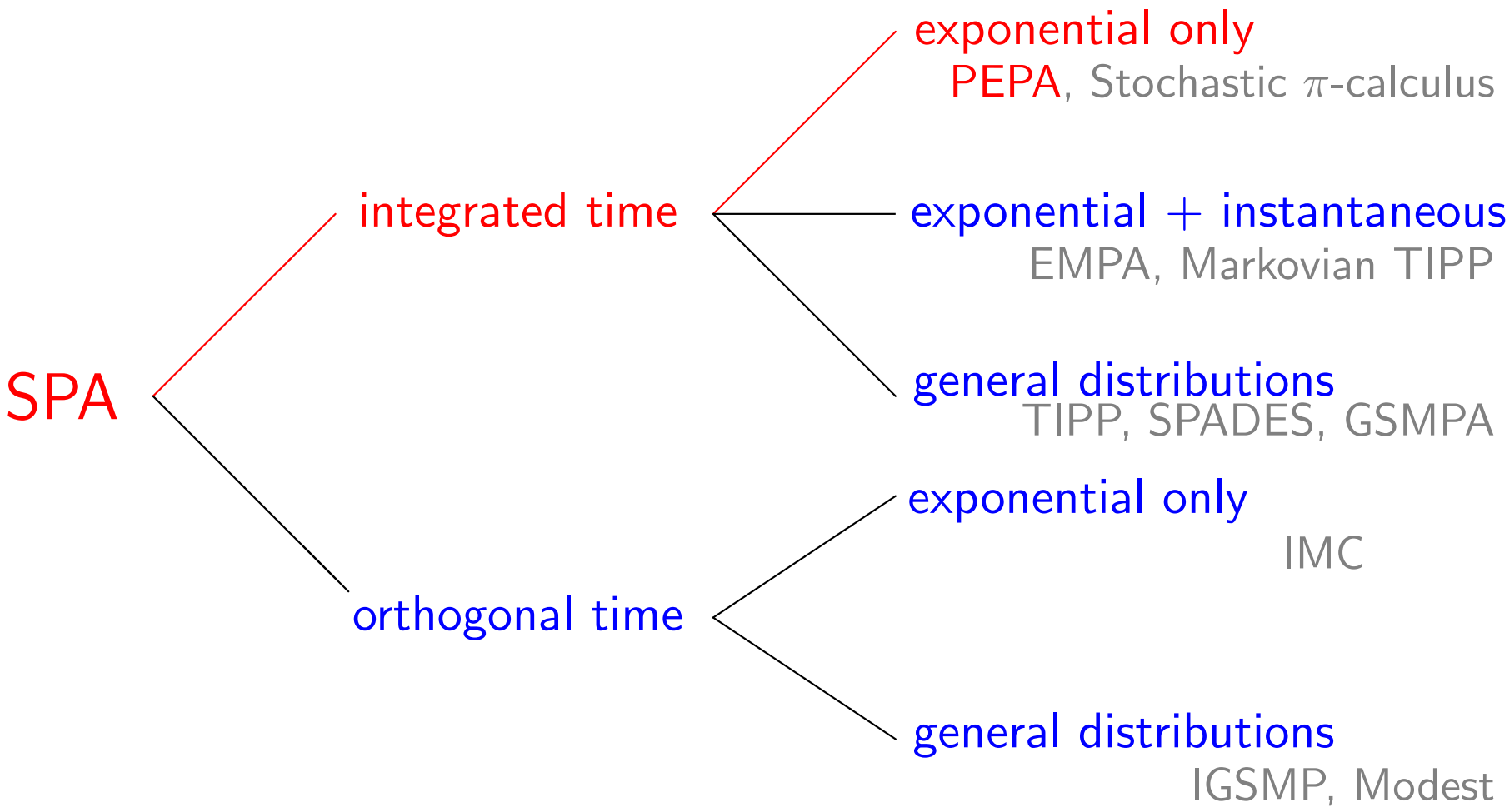
SPA Languages



SPA Languages



SPA Languages



Interplay between process algebra and Markov process

- The theoretical development underpinning PEPA has focused on the interplay between the process algebra and the underlying mathematical structure, the Markov process.

Interplay between process algebra and Markov process

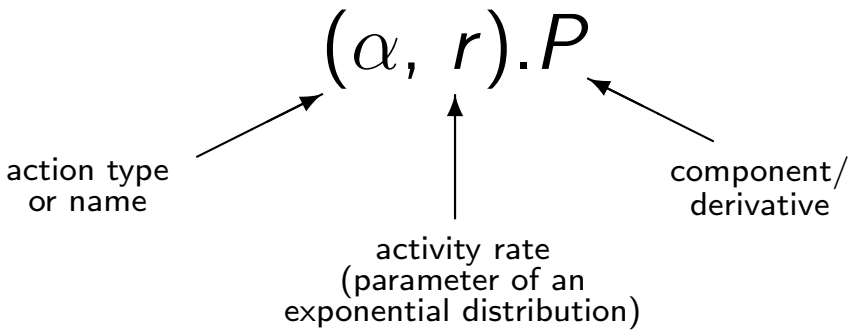
- The theoretical development underpinning PEPA has focused on the interplay between the process algebra and the underlying mathematical structure, the Markov process.
- From the process algebra side the Markov chain had a profound influence on the design of the language and in particular on the interactions between components.

Interplay between process algebra and Markov process

- The theoretical development underpinning PEPA has focused on the interplay between the process algebra and the underlying mathematical structure, the Markov process.
- From the process algebra side the Markov chain had a profound influence on the design of the language and in particular on the interactions between components.
- From the Markov chain perspective the process algebra structure has been exploited to find aspects of independence even between interacting components.

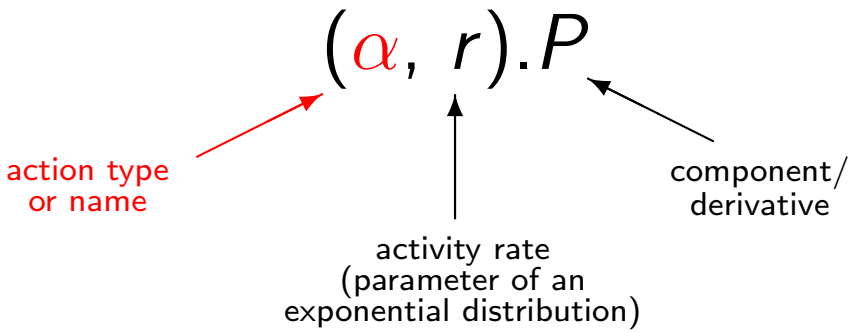
Performance Evaluation Process Algebra

- Models are constructed from components which engage in activities.



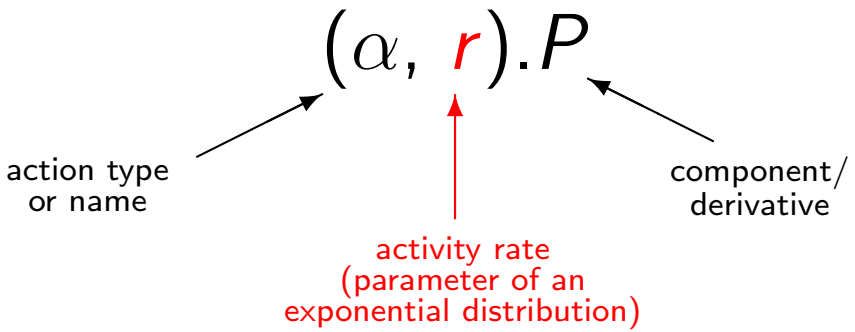
Performance Evaluation Process Algebra

- Models are constructed from components which engage in activities.



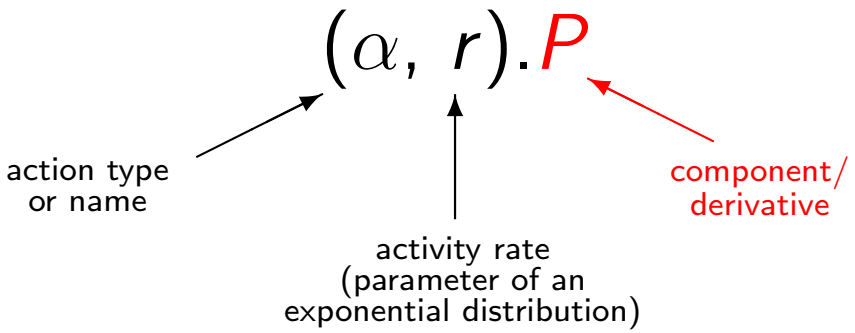
Performance Evaluation Process Algebra

- Models are constructed from components which engage in activities.



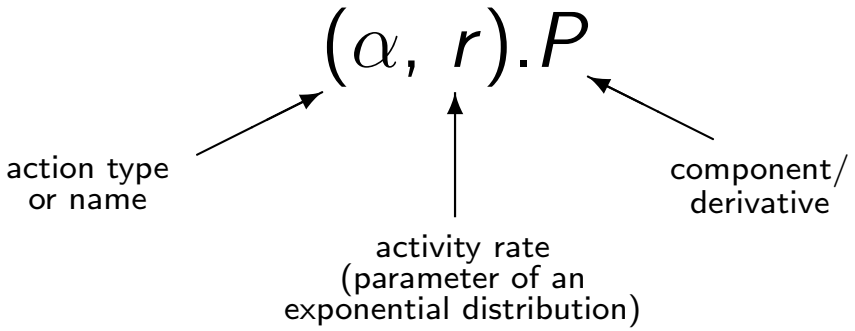
Performance Evaluation Process Algebra

- Models are constructed from components which engage in activities.



Performance Evaluation Process Algebra

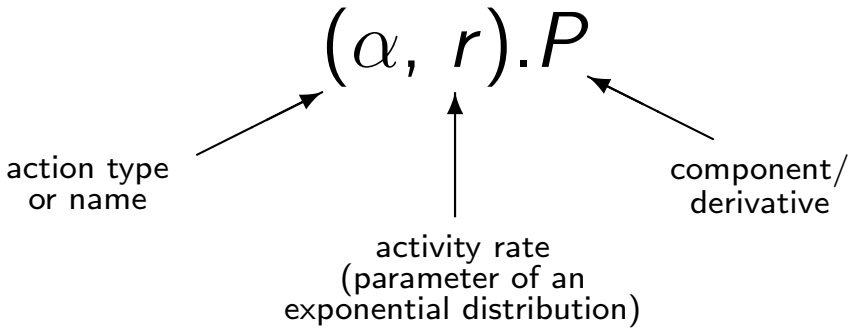
- Models are constructed from components which engage in activities.



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

Performance Evaluation Process Algebra

- Models are constructed from components which engage in activities.

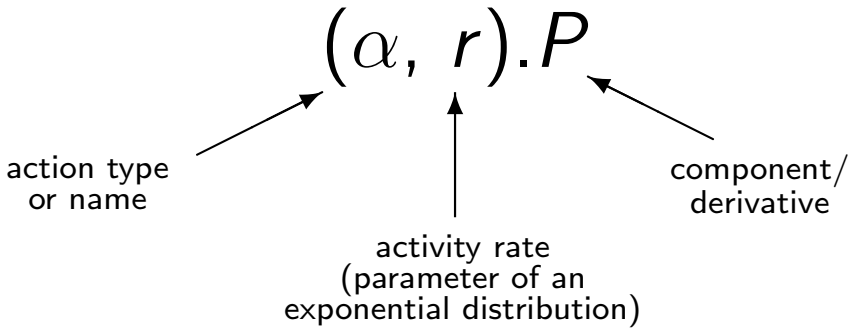


- The language is used to generate a CTMC for performance modelling.

PEPA
MODEL

Performance Evaluation Process Algebra

- Models are constructed from components which engage in activities.

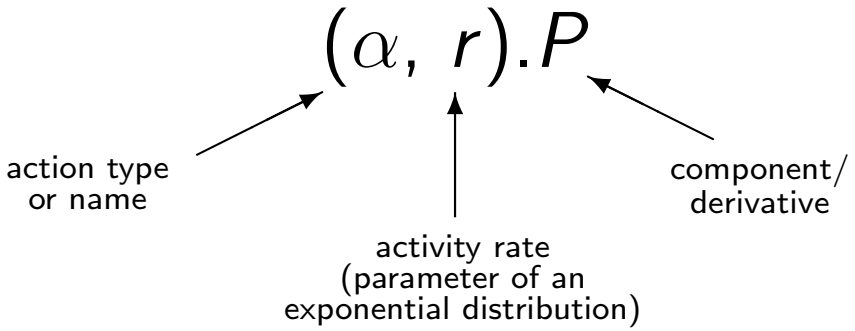


- The language is used to generate a CTMC for performance modelling.

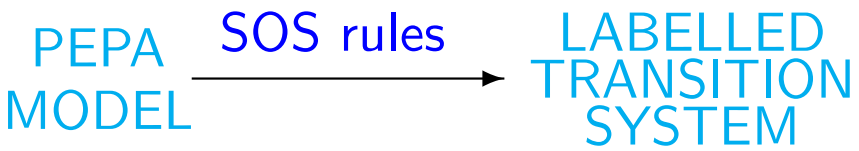
PEPA MODEL $\xrightarrow{\text{SOS rules}}$

Performance Evaluation Process Algebra

- Models are constructed from components which engage in activities.

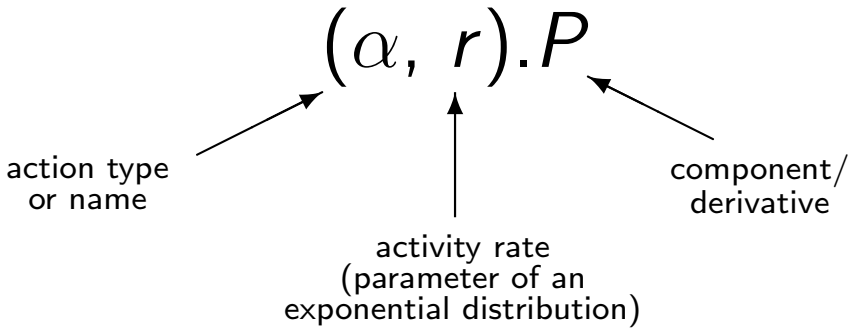


- The language is used to generate a CTMC for performance modelling.



Performance Evaluation Process Algebra

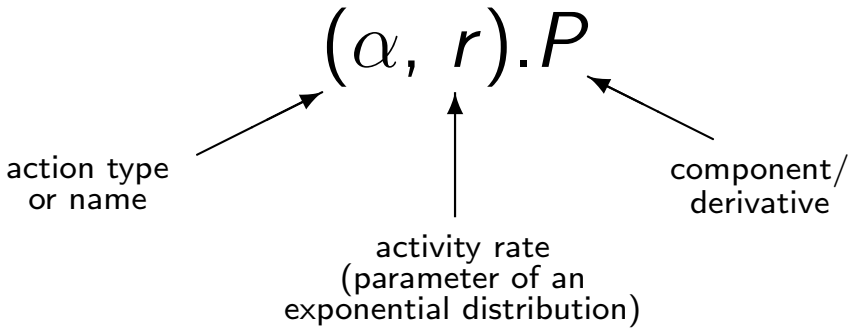
- Models are constructed from components which engage in activities.



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

Performance Evaluation Process Algebra

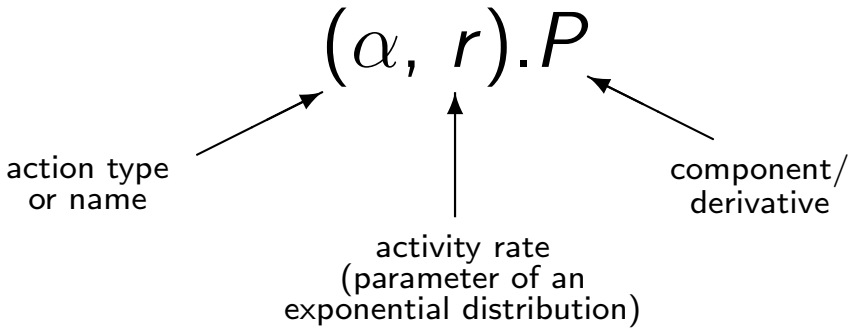
- Models are constructed from components which engage in activities.



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

Performance Evaluation Process Algebra

- Models are constructed from components which engage in activities.



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

PEPA

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

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

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

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
CONSTANT:	$A \stackrel{def}{=} S$	assigning names

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
CONSTANT:	$A \stackrel{def}{=} S$	assigning names
COOPERATION:	$P \underset{L}{\bowtie} P$	$\alpha \notin L$ individual actions
		$\alpha \in L$ 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
CONSTANT:	$A \stackrel{def}{=} S$	assigning names
COOPERATION:	$P \underset{L}{\bowtie} P$	$\alpha \notin L$ individual actions $\alpha \in L$ shared actions
HIDING:	P/L	abstraction $\alpha \in L \Rightarrow \alpha \rightarrow \tau$

Example: Browsers, server and download

$$
\begin{aligned}
\mathit{Server} &\stackrel{\mathit{def}}{=} (\mathit{get}, \top).(\mathit{download}, \mu).(\mathit{rel}, \top).\mathit{Server} \\
\mathit{Browser} &\stackrel{\mathit{def}}{=} (\mathit{display}, p\lambda).(\mathit{get}, g).(\mathit{download}, \top).(\mathit{rel}, r).\mathit{Browser} \\
&\quad + \; (\mathit{display}, (1-p)\lambda).(\mathit{cache}, m).\mathit{Browser} \\
\mathit{WEB} &\stackrel{\mathit{def}}{=} \left(\mathit{Browser} \parallel \mathit{Browser}\right) \underset{L}{\bowtie} \mathit{Server}
\end{aligned}
$$

where $L = \{\mathit{get}, \mathit{download}, \mathit{rel}\}$

Structured Operational Semantics

PEPA is defined using a Plotkin-style structured operational semantics (a "small step" semantics).

Structured Operational Semantics

PEPA is defined using a Plotkin-style structured operational semantics (a "small step" semantics).

Prefix

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

Structured Operational Semantics

PEPA is defined using a Plotkin-style structured operational semantics (a "small step" semantics).

Prefix

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

Choice

$$\frac{E \xrightarrow{(\alpha,r)} E'}{E + F \xrightarrow{(\alpha,r)} E'}$$

$$\frac{F \xrightarrow{(\alpha,r)} F'}{E + F \xrightarrow{(\alpha,r)} F'}$$

Structured Operational Semantics: Cooperation ($\alpha \notin L$)

Cooperation

$$\frac{E \xrightarrow{(\alpha,r)} E'}{E \underset{L}{\bowtie} F \xrightarrow{(\alpha,r)} E' \underset{L}{\bowtie} F} \ (\alpha \notin L)$$

Structured Operational Semantics: Cooperation ($\alpha \notin L$)

Cooperation

$$\frac{E \xrightarrow{(\alpha,r)} E'}{E \underset{L}{\bowtie} F \xrightarrow{(\alpha,r)} E' \underset{L}{\bowtie} F} \ (\alpha \notin L)$$

$$\frac{F \xrightarrow{(\alpha,r)} F'}{E \underset{L}{\bowtie} F \xrightarrow{(\alpha,r)} E \underset{L}{\bowtie} F'} \ (\alpha \notin L)$$

Structured Operational Semantics: Cooperation ($\alpha \in L$)

Cooperation

$$\frac{E \xrightarrow{(\alpha, r_1)} E' \qquad F \xrightarrow{(\alpha, r_2)} F'}{E \underset{L}{\bowtie} F \xrightarrow{(\alpha, R)} E' \underset{L}{\bowtie} F'} \ (\alpha \in L)$$

Structured Operational Semantics: Cooperation ($\alpha \in L$)

Cooperation

$$\frac{E \xrightarrow{(\alpha, r_1)} E' \qquad F \xrightarrow{(\alpha, r_2)} F'}{E \underset{L}{\bowtie} F \xrightarrow{(\alpha, R)} E' \underset{L}{\bowtie} F'} \ (\alpha \in L)$$

where $R = \dfrac{r_1}{r_\alpha(E)} \dfrac{r_2}{r_\alpha(F)} min(r_\alpha(E), r_\alpha(F))$

Apparent Rate

$$
\begin{aligned}
r_\alpha((\beta, r).P) &= \begin{cases} r & \beta = \alpha \\ 0 & \beta \neq \alpha \end{cases} \\
r_\alpha(P + Q) &= r_\alpha(P) + r_\alpha(Q) \\
r_\alpha(A) &= r_\alpha(P) \qquad \text{where } A \stackrel{def}{=} P \\
r_\alpha(P \underset{L}{\bowtie} Q) &= \begin{cases} r_\alpha(P) + r_\alpha(Q) & \alpha \notin L \\ min(r_\alpha(P), r_\alpha(Q)) & \alpha \in L \end{cases} \\
r_\alpha(P/L) &= \begin{cases} r_\alpha(P) & \alpha \notin L \\ 0 & \alpha \in L \end{cases}
\end{aligned}
$$

Structured Operational Semantics: Hiding

Hiding

$$\frac{E \xrightarrow{(\alpha,r)} E'}{E/L \xrightarrow{(\alpha,r)} E'/L} \; (\alpha \notin L)$$

Structured Operational Semantics: Hiding

Hiding

$$\frac{E \xrightarrow{(\alpha,r)} E'}{E/L \xrightarrow{(\alpha,r)} E'/L} \ (\alpha \notin L)$$

$$\frac{E \xrightarrow{(\alpha,r)} E'}{E/L \xrightarrow{(\tau,r)} E'/L} \ (\alpha \in L)$$

Structured Operational Semantics: Constants

Constant

$$\frac{E \xrightarrow{(\alpha,r)} E'}{A \xrightarrow{(\alpha,r)} E'} \, (A \stackrel{def}{=} E)$$

Properties of the definition (1)

PEPA has no "nil" (a deadlocked process).

This is because the PEPA language is intended for modelling non-stop processes (such as Web servers, operating systems, or manufacturing processes) rather than for modelling terminating processes (a compilation, a sorting routine, and so forth).

Creating a deadlocked process

When we are interested in transient behaviour we use the deadlocked process *Stop* to signal a component which performs no further actions.

$$Stop \stackrel{def}{=} \Big(\big((a, r).Stop \big) \underset{\{a,b\}}{\bowtie} \big((b, r).Stop \big) \Big) /\{ a, b \}$$

Properties of the definition (2)

Cooperation in PEPA is multi-way. Two, three, four or more partners may cooperate, and they all need to synchronise for the activity to happen.

Properties of the definition (2)

Cooperation in PEPA is multi-way. Two, three, four or more partners may cooperate, and they all need to synchronise for the activity to happen.

This comes from the fact that synchronisation has the form $a, a \rightarrow a$ (as in CSP) instead of $a, \bar{a} \rightarrow \tau$ (as in CCS and the π-calculus).

Properties of the definition (2)

Cooperation in PEPA is multi-way. Two, three, four or more partners may cooperate, and they all need to synchronise for the activity to happen.

This comes from the fact that synchronisation has the form $a, a \rightarrow a$ (as in CSP) instead of $a, \bar{a} \rightarrow \tau$ (as in CCS and the π-calculus).

This is used to have "witnesses" to events (known as stochastic probes).

Properties of the definition (3)

- Because of its mapping onto a CTMC, PEPA has an interleaving semantics.

Properties of the definition (3)

- Because of its mapping onto a CTMC, PEPA has an interleaving semantics.
- Other modelling formalisms based on CTMCs are also based on an interleaving semantics (e.g. Generalised Stochastic Petri nets).

Properties of the definition (3)

- Because of its mapping onto a CTMC, PEPA has an interleaving semantics.
- Other modelling formalisms based on CTMCs are also based on an interleaving semantics (e.g. Generalised Stochastic Petri nets).
- As we have seen a continuous time Markov chain (CTMC) is generated from a PEPA model via its structured operational semantics.

Properties of the definition (3)

- Because of its mapping onto a CTMC, PEPA has an interleaving semantics.
- Other modelling formalisms based on CTMCs are also based on an interleaving semantics (e.g. Generalised Stochastic Petri nets).
- As we have seen a continuous time Markov chain (CTMC) is generated from a PEPA model via its structured operational semantics.
- Linear algebra is used to solve the model in terms of equilibrium behaviour.

Properties of the definition (3)

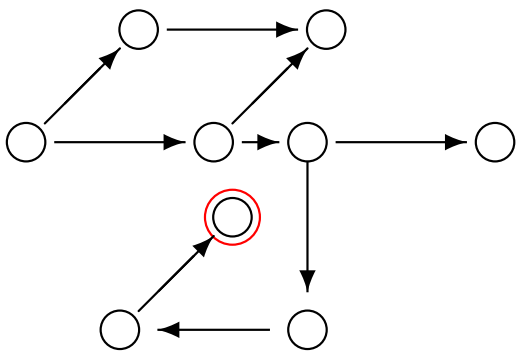
- Because of its mapping onto a CTMC, PEPA has an interleaving semantics.
- Other modelling formalisms based on CTMCs are also based on an interleaving semantics (e.g. Generalised Stochastic Petri nets).
- As we have seen a continuous time Markov chain (CTMC) is generated from a PEPA model via its structured operational semantics.
- Linear algebra is used to solve the model in terms of equilibrium behaviour.
- The resulting probability distribution is seldom the ultimate goal of performance analysis; a modeller derives performance measures from this distribution via a reward structure.

Integrated analysis

Qualitative verification can now be complemented by quantitative verification.

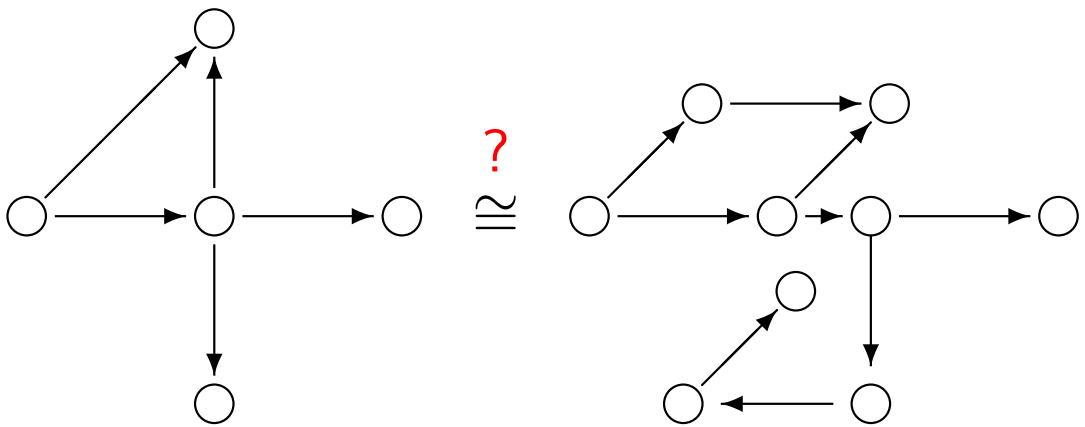
Integrated analysis: Reachability analysis

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



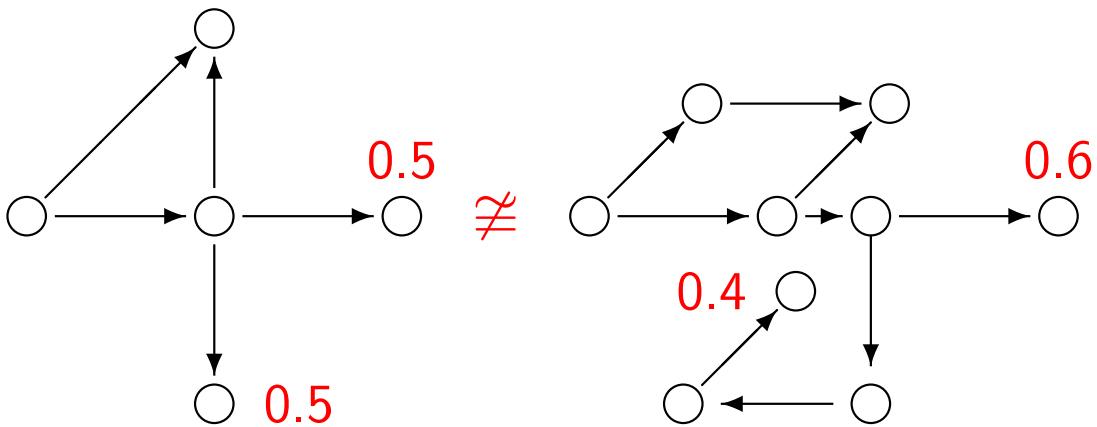
Integrated analysis: Specification matching

With what probability
does system behaviour
match its specification?



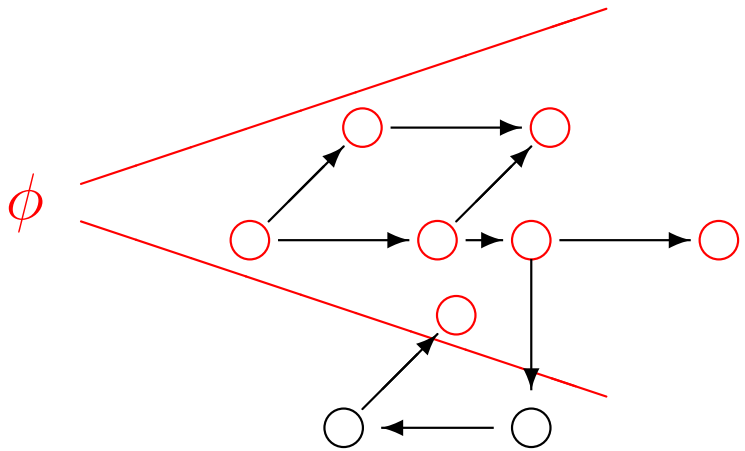
Integrated analysis: Specification matching

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



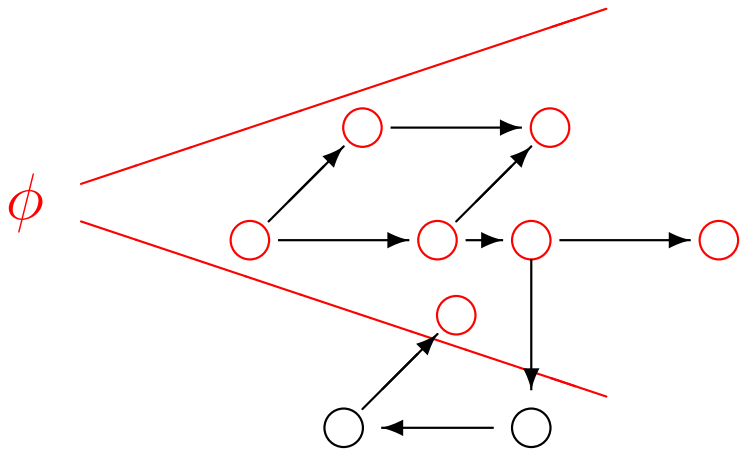
Integrated analysis: Model checking

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

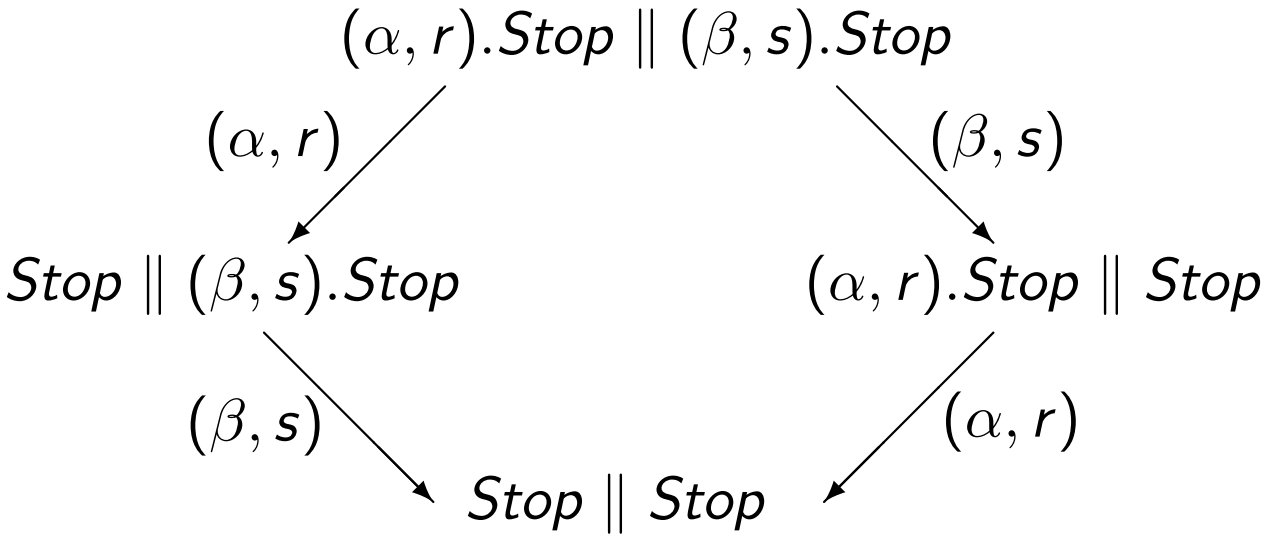


Integrated analysis: Model checking

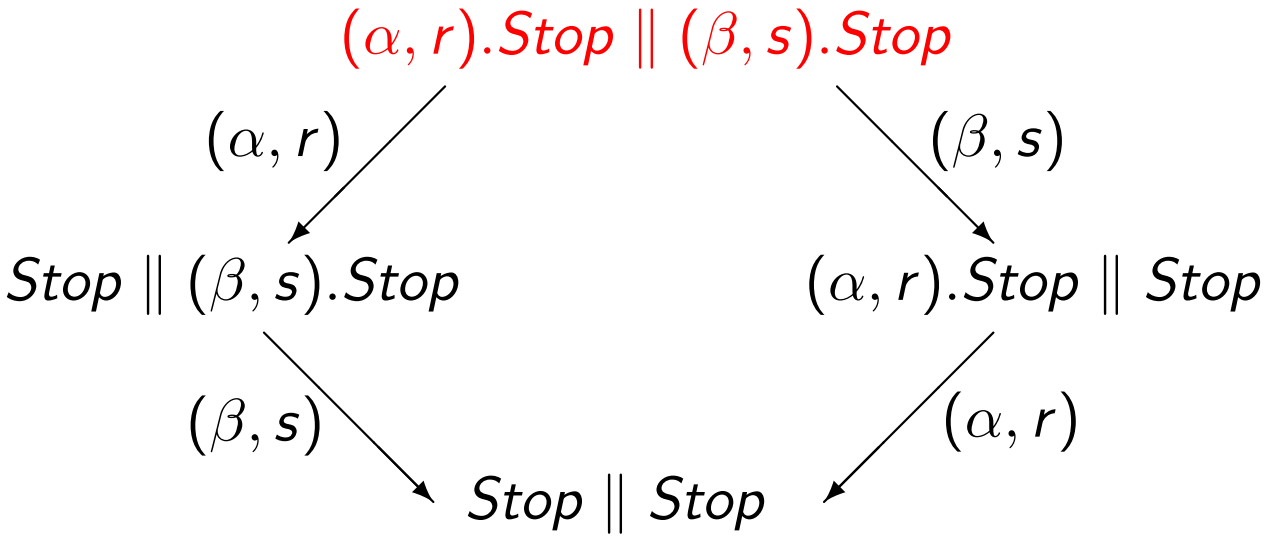
For a given starting state
how long is it until
a given property ϕ holds?



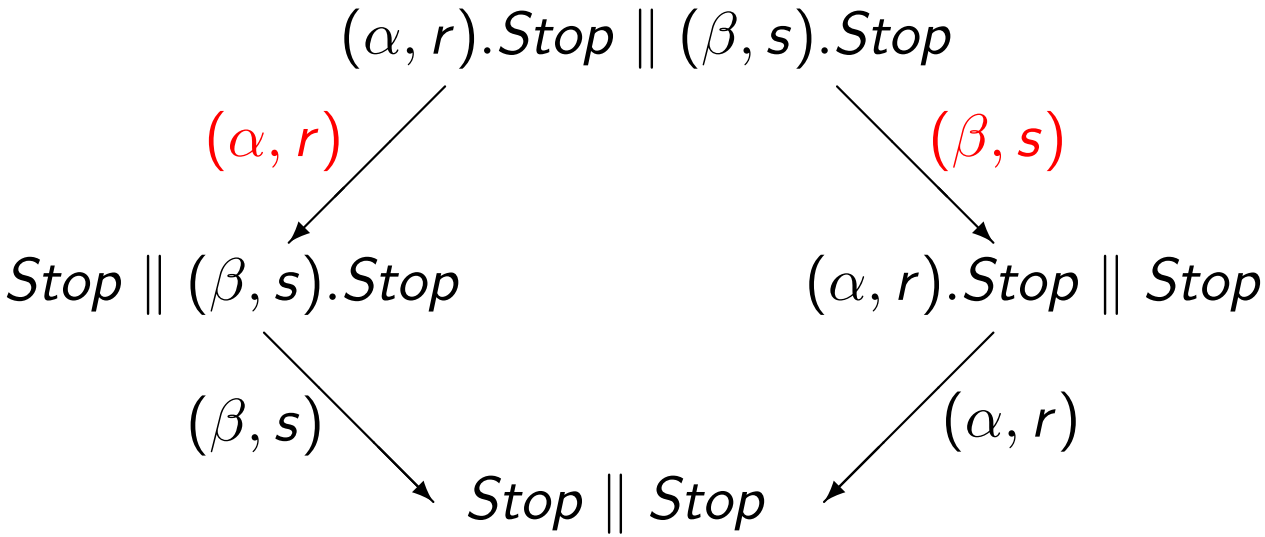
The Importance of Being Exponential



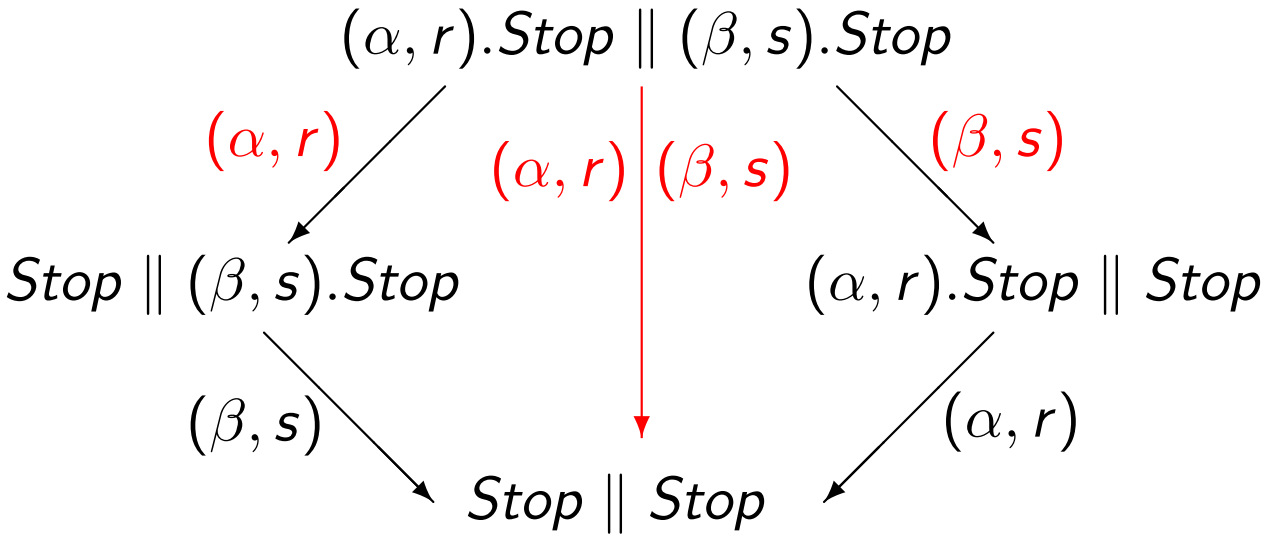
The Importance of Being Exponential



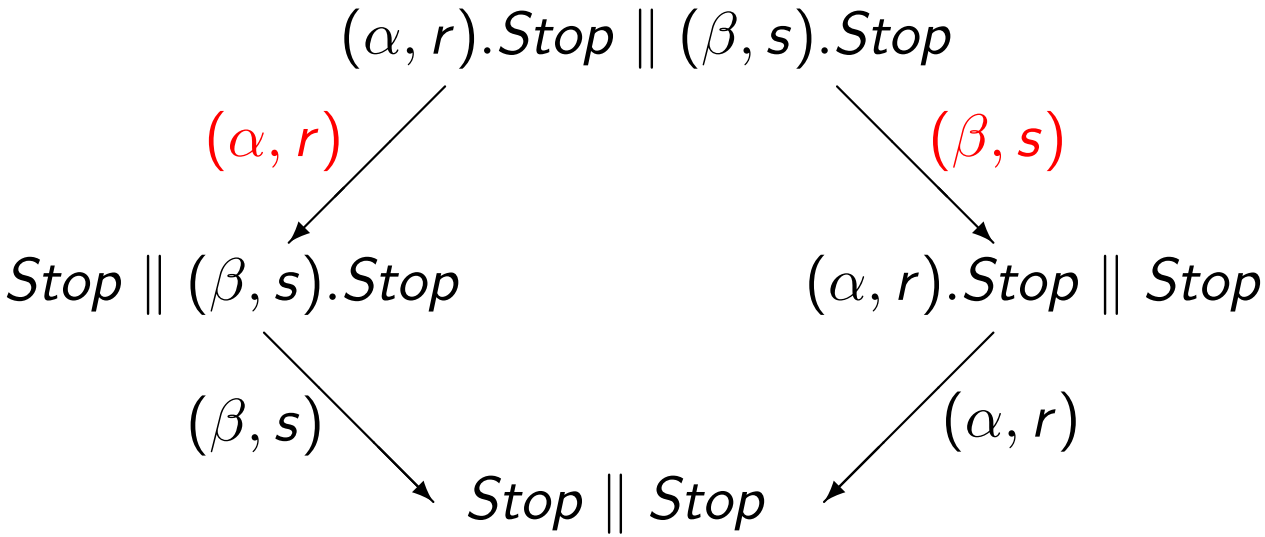
The Importance of Being Exponential



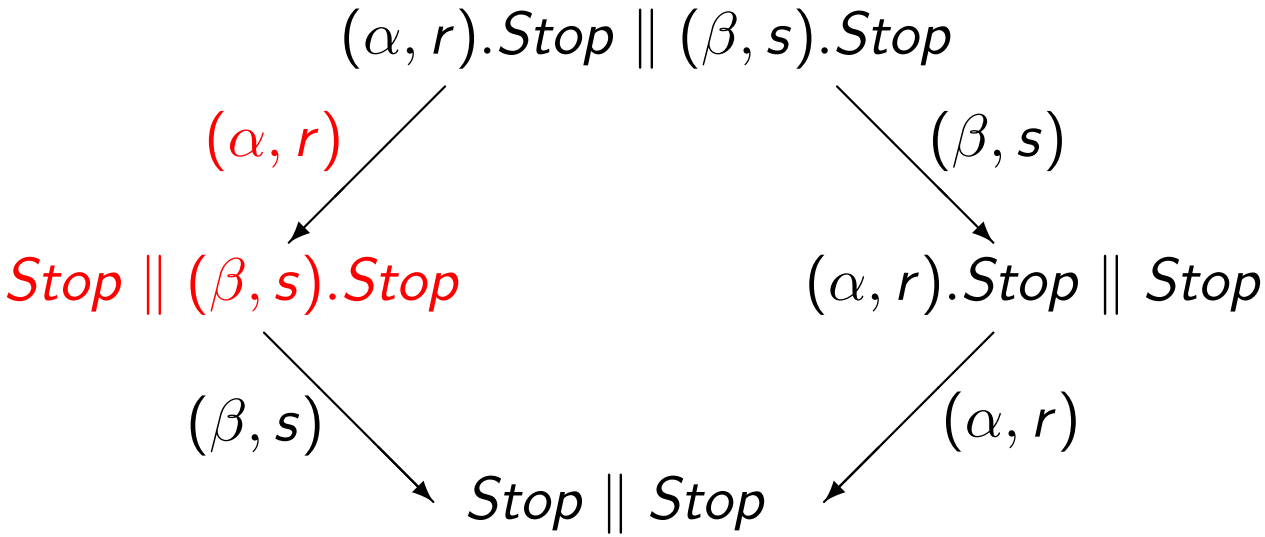
The Importance of Being Exponential



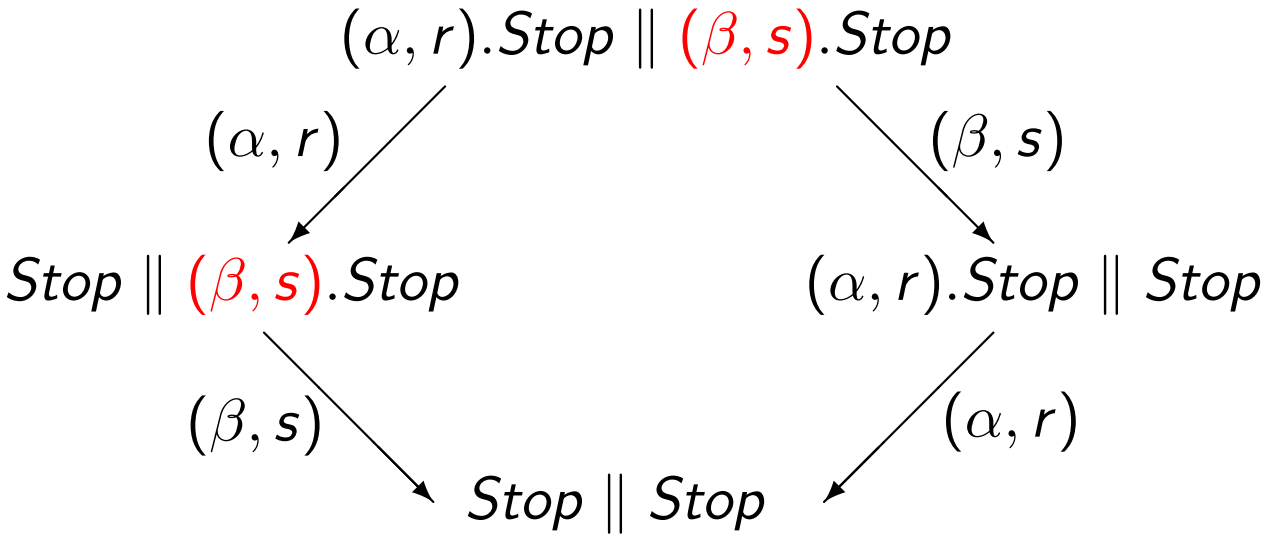
The Importance of Being Exponential



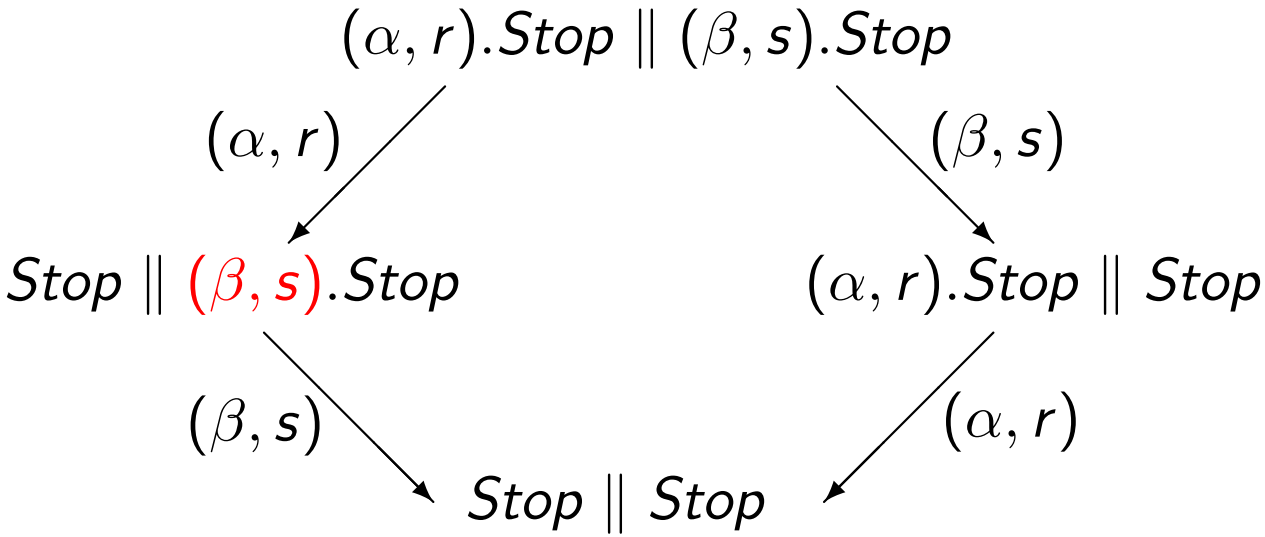
The Importance of Being Exponential



The Importance of Being Exponential



The Importance of Being Exponential



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

The exponential distribution and the expansion law

We retain the expansion law of classical process algebra:

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

only if the negative exponential distribution is used.

Outline

Parallel Composition

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

Parallel Composition

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

Parallel Composition

- Parallel 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

- Parallel 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.

Parallel Composition

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

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.
- 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.
- 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

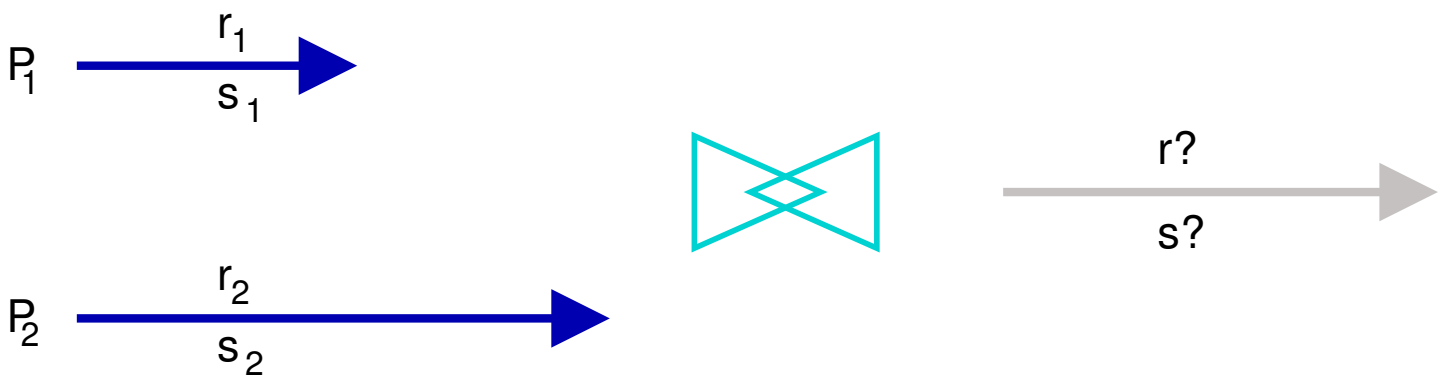
- Actions are partitioned into input and output pairs.
- Communication or synchronisation takes places between conjugate pairs.
- The resulting action has silent type τ.

CSP-style

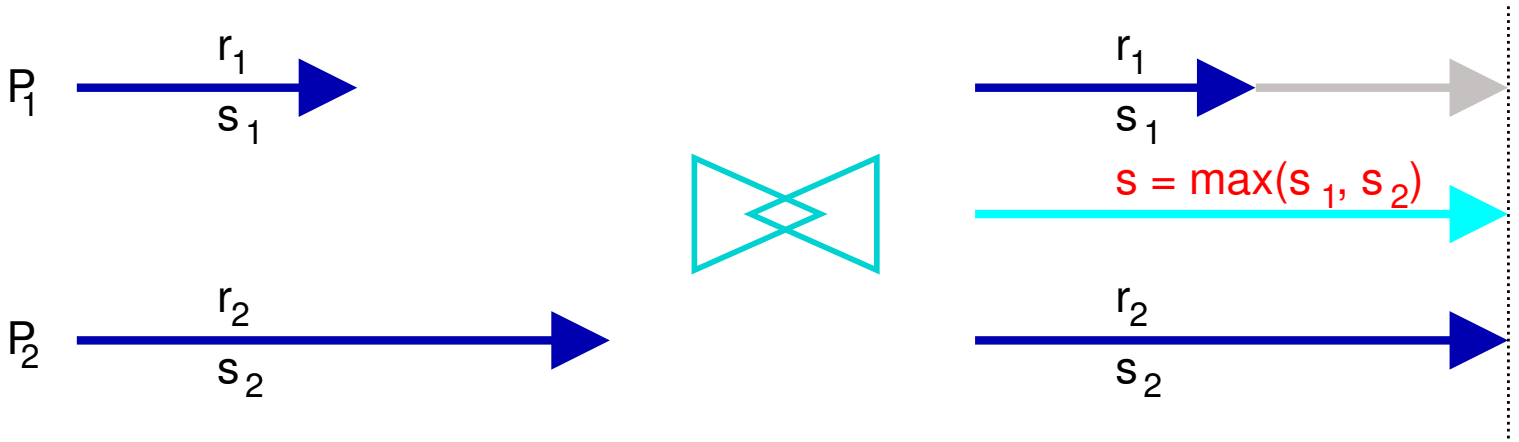
- No distinction between input and output actions.
- Communication or synchronisation takes place on the basis of shared names.
- The resulting action has the same name.

Most stochastic process algebras adopt CSP-style synchronisation.

Timed Synchronisation

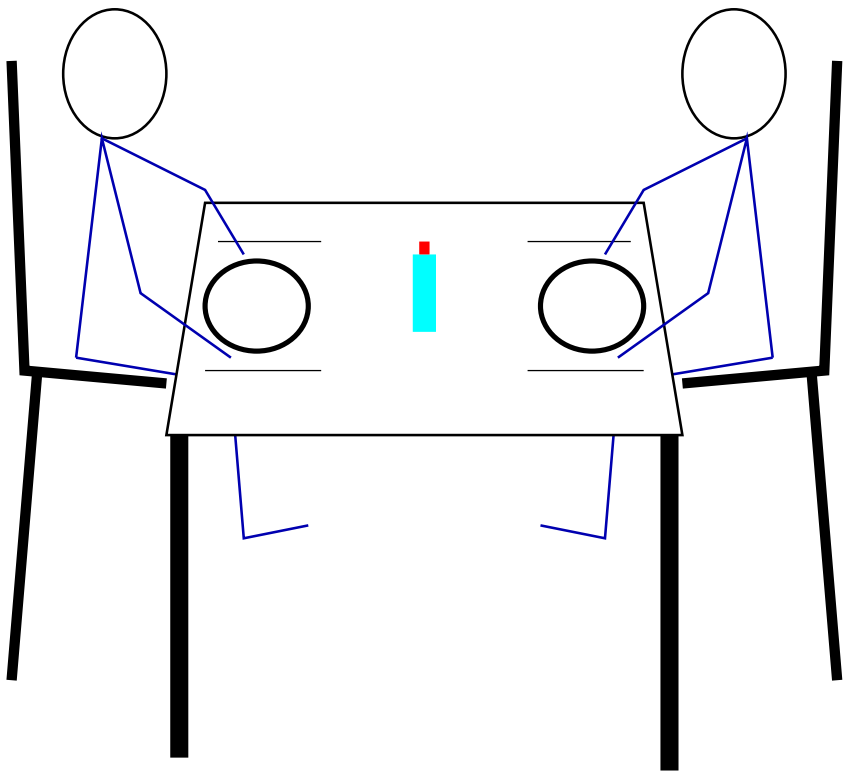


Timed Synchronisation

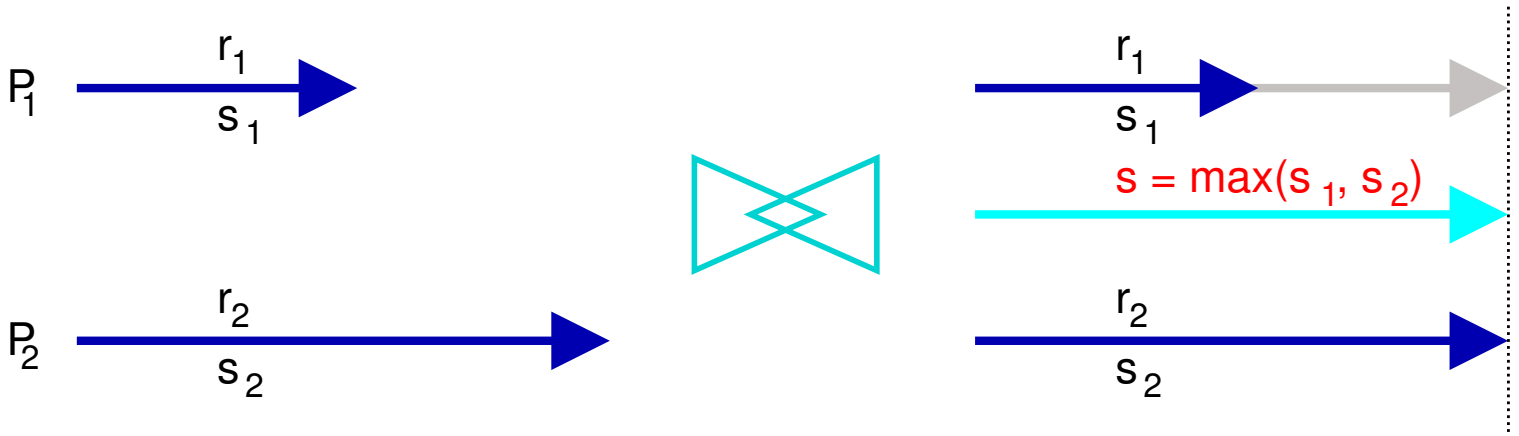


Barrier Synchronisation

Timed Synchronisation

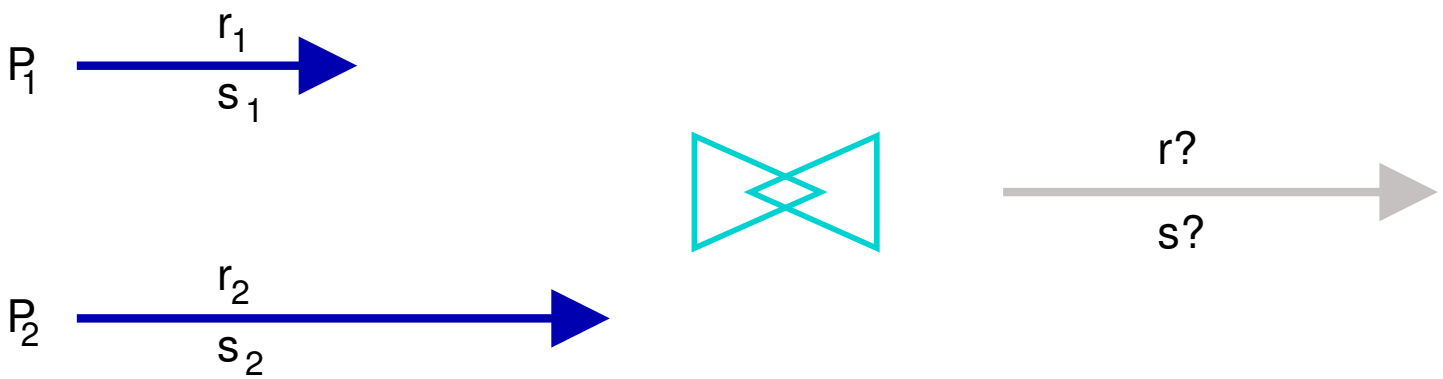


Timed Synchronisation



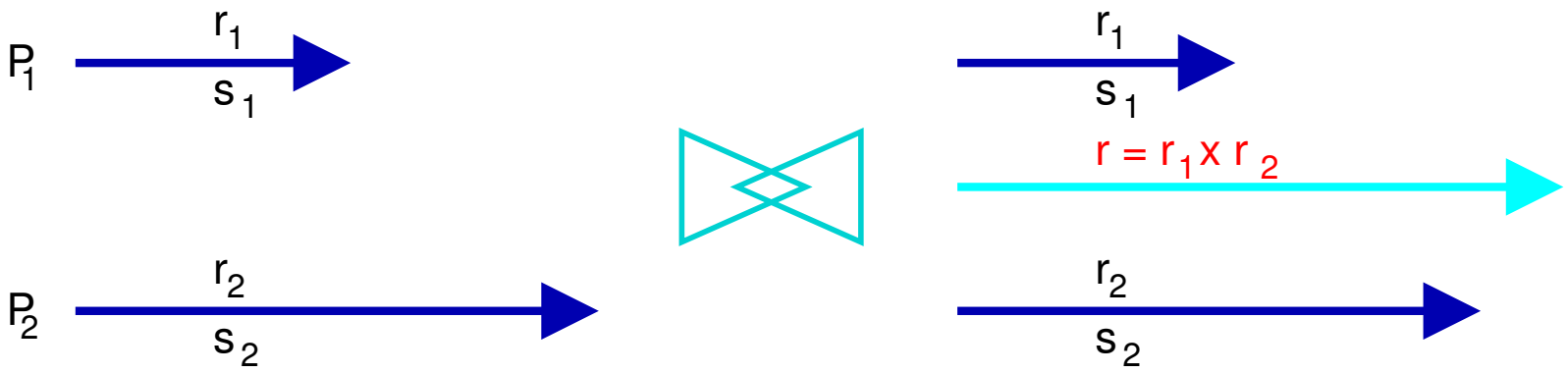
s is no longer exponentially distributed

Timed Synchronisation



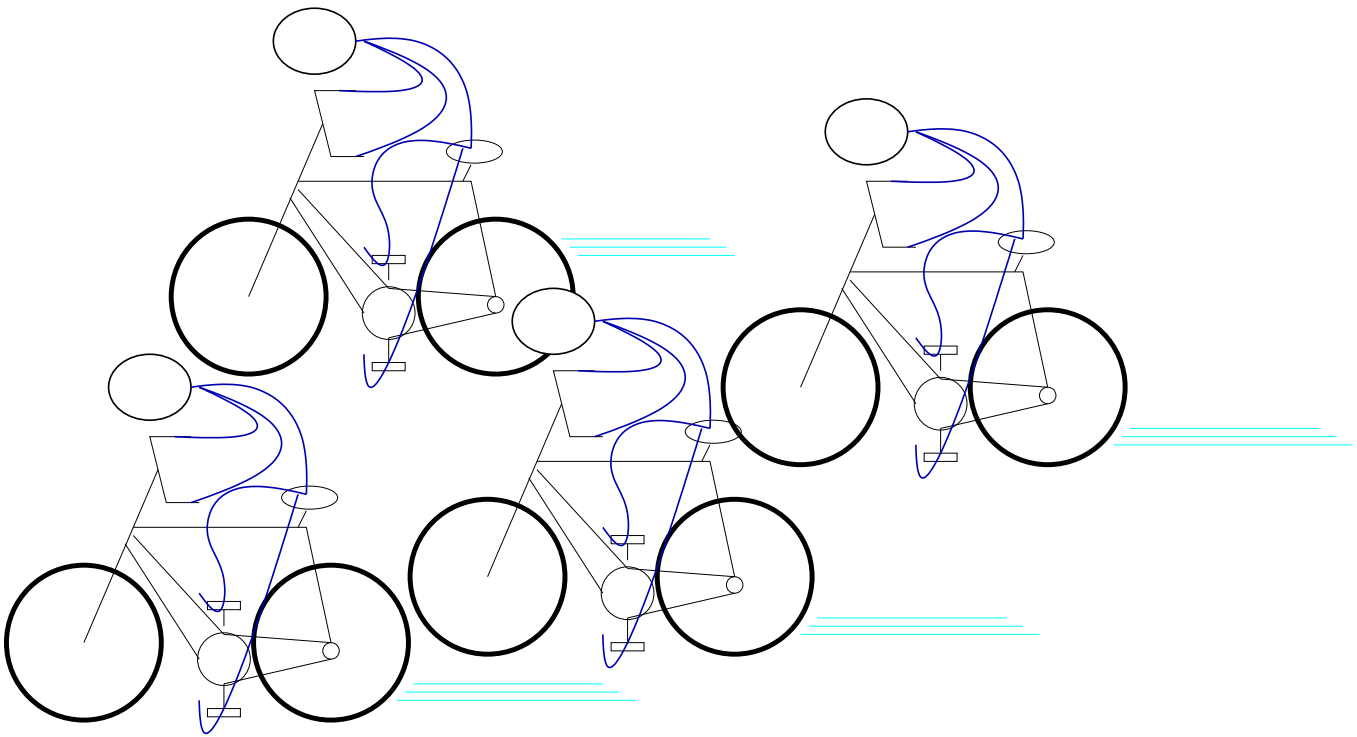
algebraic considerations limit choices

Timed Synchronisation

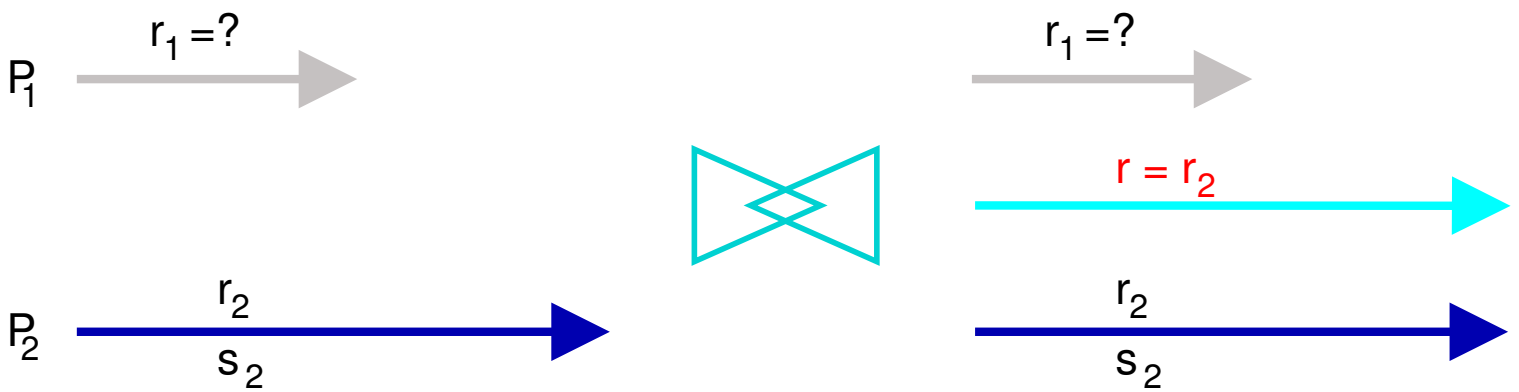


TIPP: new rate is product of individual rates

Timed Synchronisation

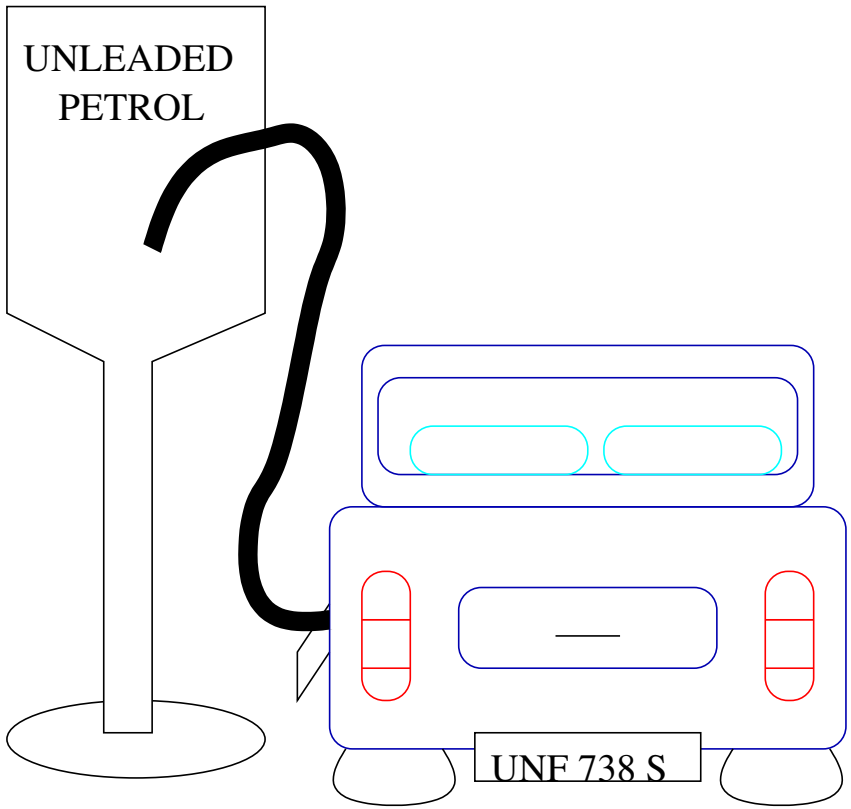


Timed Synchronisation

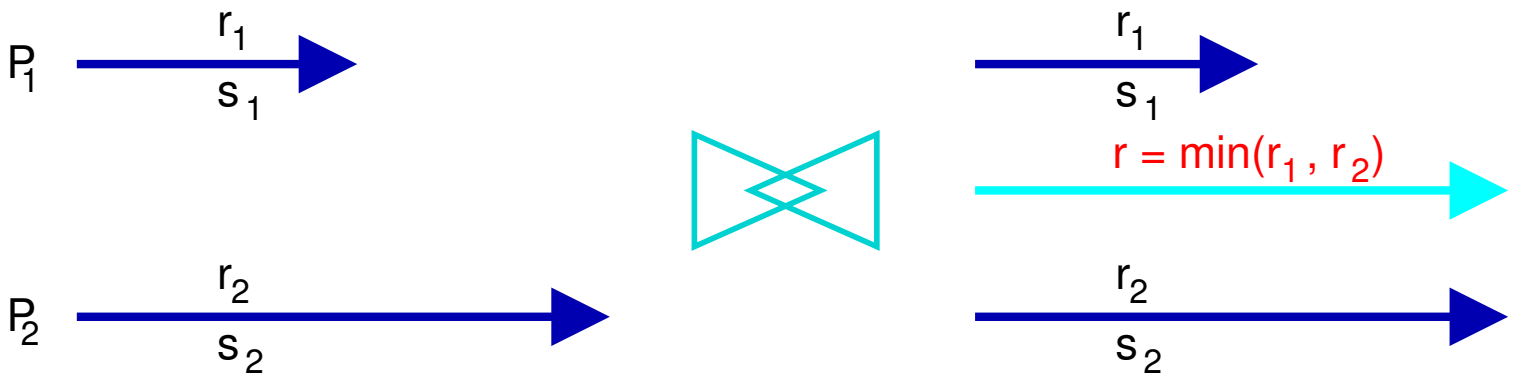


EMPA: one participant is passive

Timed Synchronisation

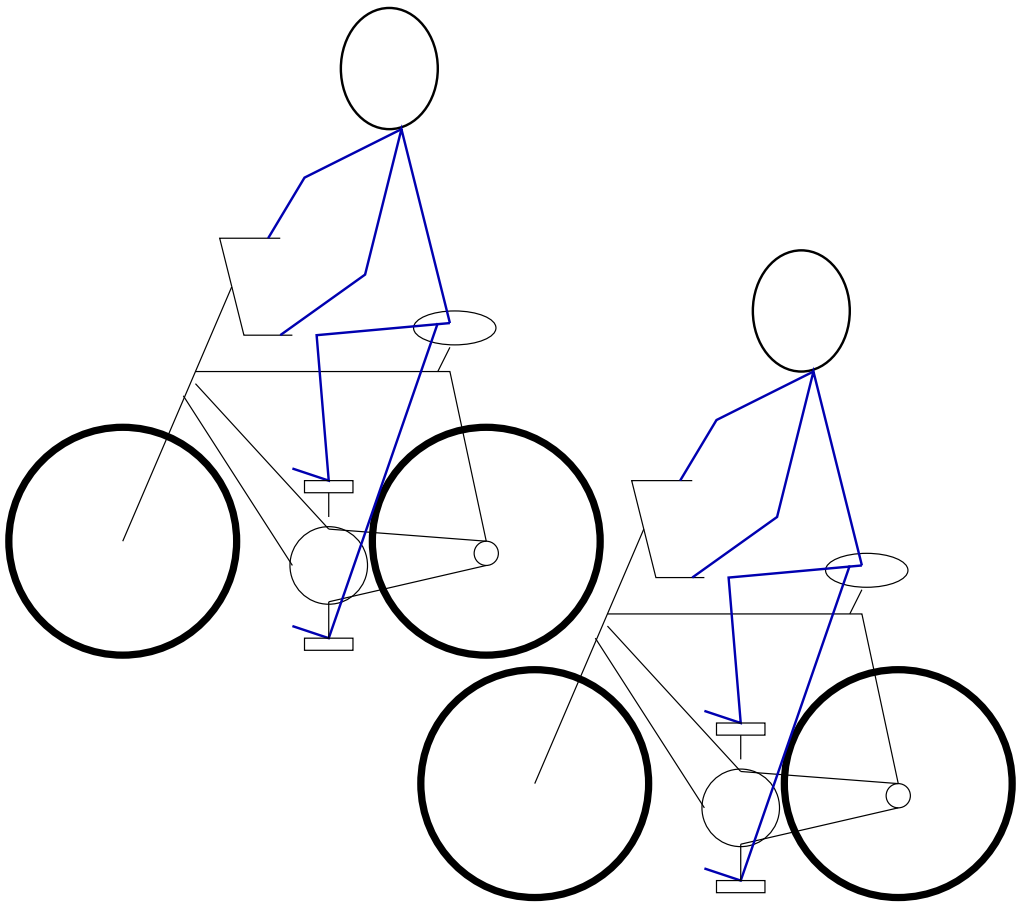


Timed Synchronisation



bounded capacity: new rate is the minimum of the rates

Timed Synchronisation



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

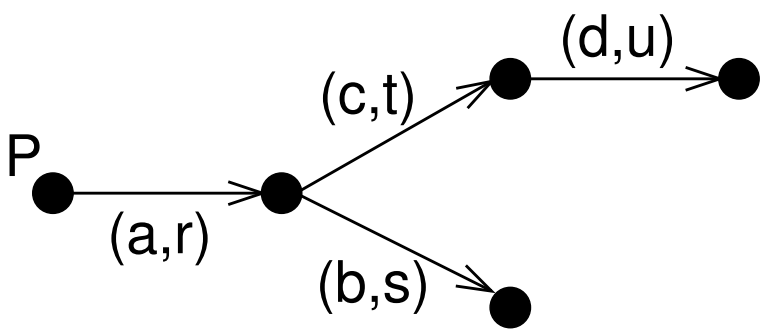
- In PEPA each component has a bounded capacity to carry out activities of any particular type, determined by the apparent rate for that type.
- Synchronisation, or cooperation cannot make a component exceed its bounded capacity.
- Thus the apparent rate of a cooperation is the minimum of the apparent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

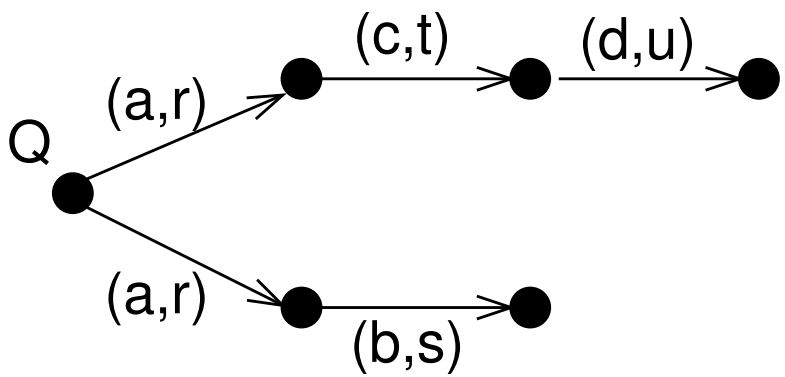
Outline

Equivalence Relations

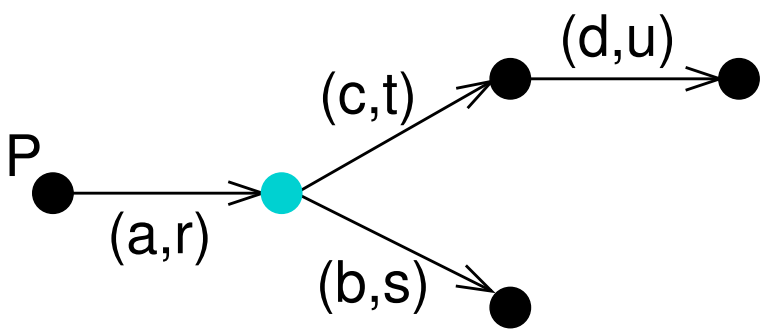
As in CCS-style process algebra, equivalence relations are defined based on the notion of observability.

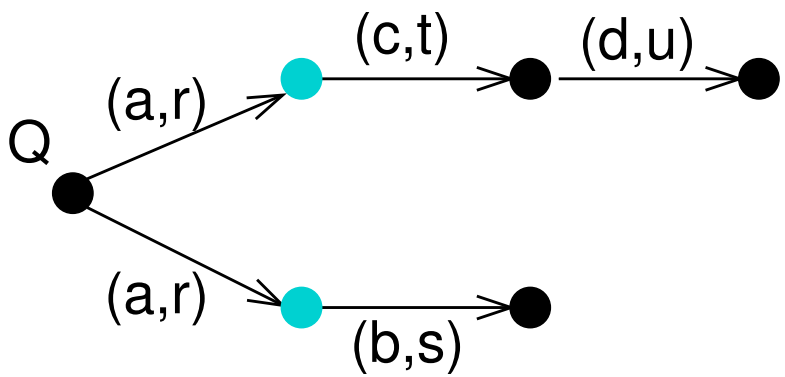
Equivalence Relations



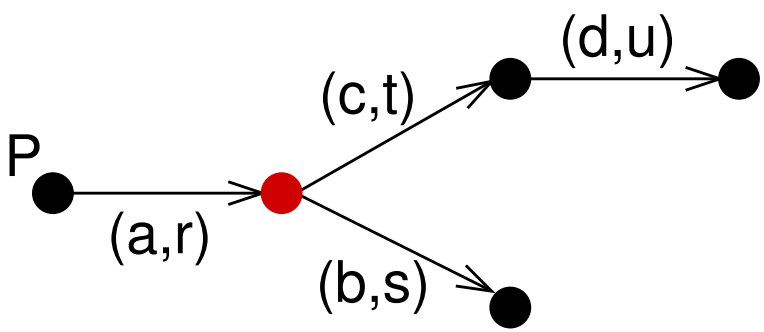


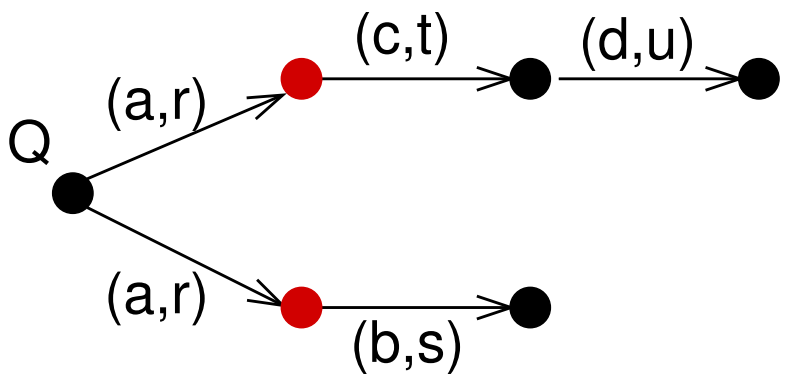
Equivalence Relations





Equivalence Relations

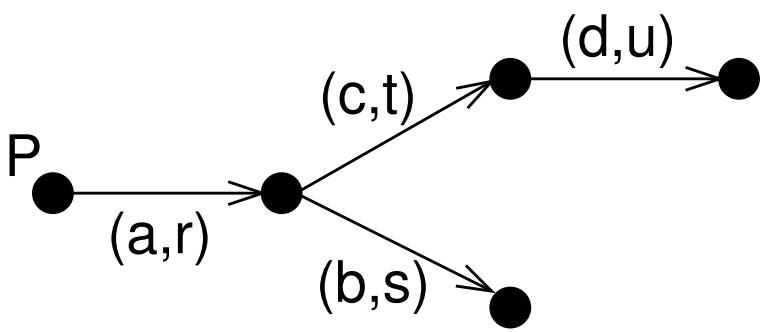


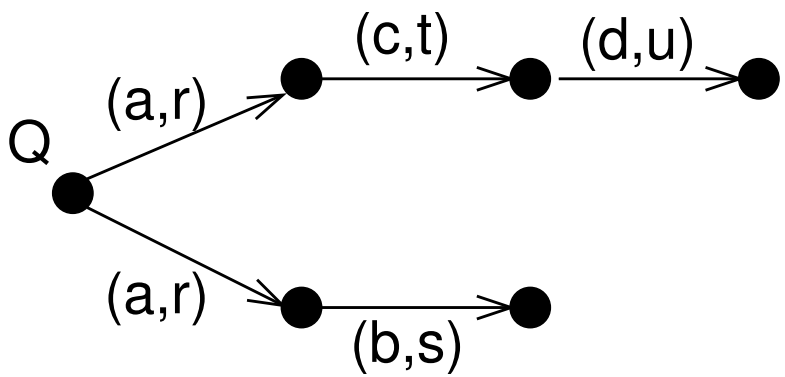


Equivalence Relations

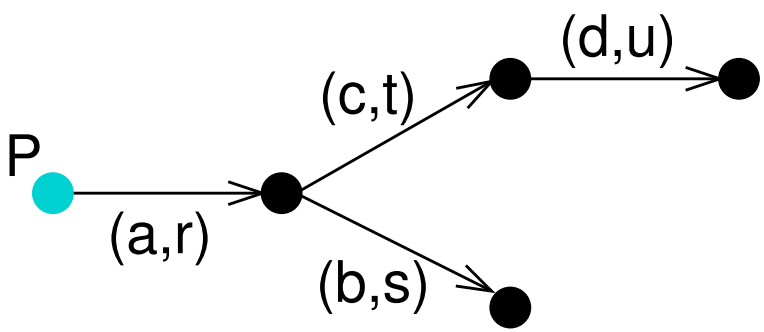
In PEPA observation is assumed to include the ability to record timing information over a number of runs.

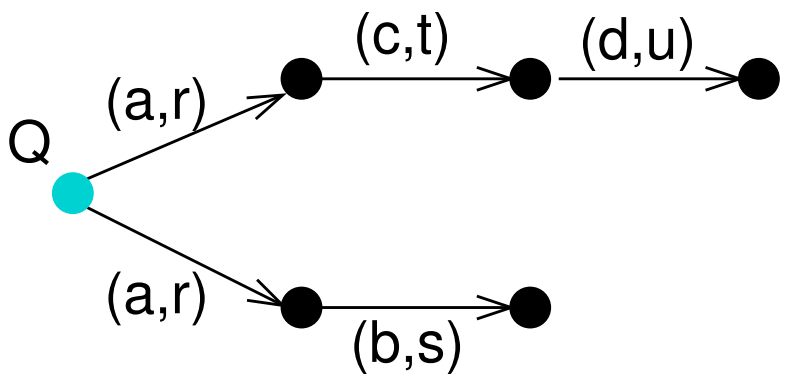
Equivalence Relations



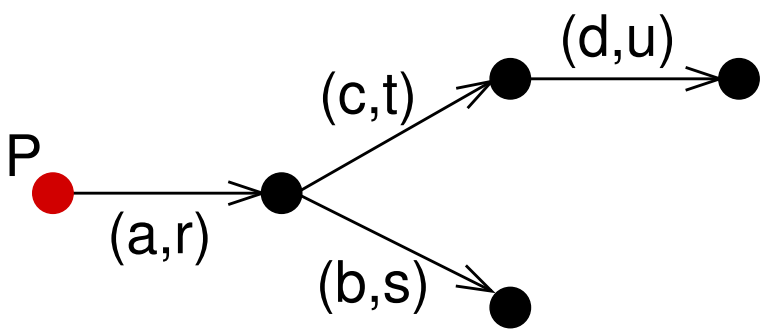


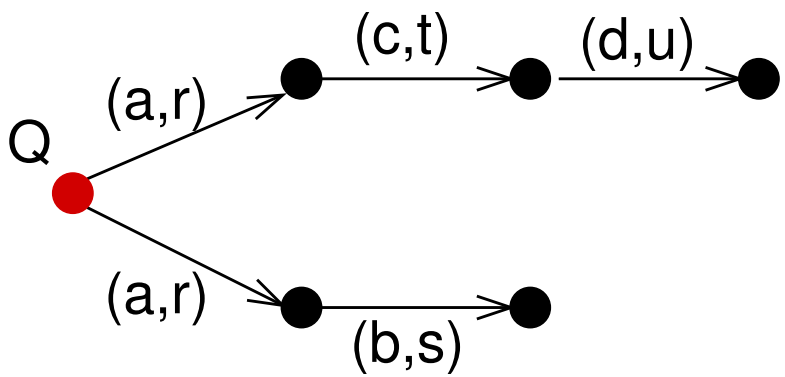
Equivalence Relations



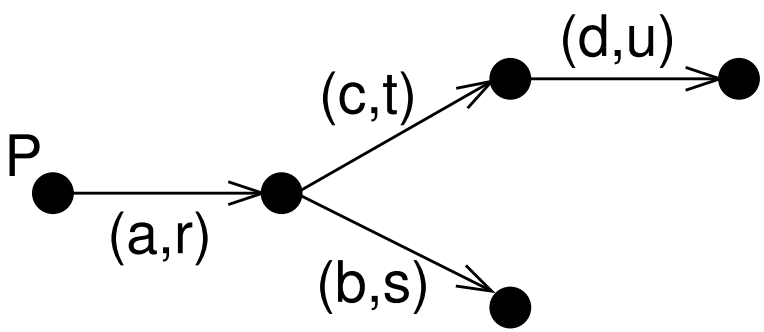


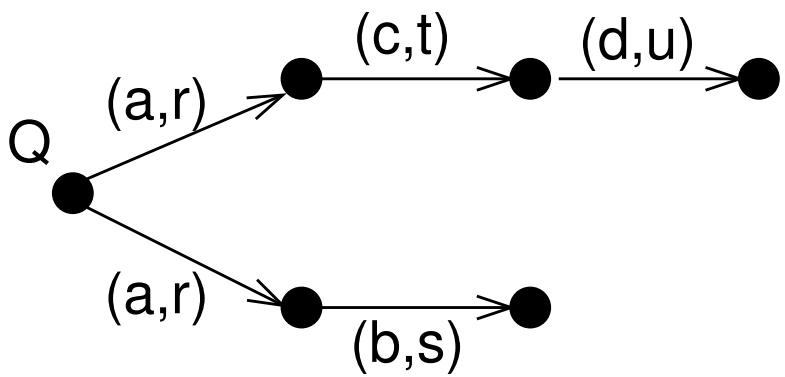
Equivalence Relations





Equivalence Relations





Equivalence Relations

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

Equivalence Relations

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

Moreover this bisimulation is a congruence for all the combinators of PEPA.

Equivalence Relations

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

Moreover this bisimulation is a congruence for all the combinators of PEPA.

The consequences of the relationship between Markovian bisimulation and lumpability and the property that it is a congruence will be discussed in more detail in Lecture 4.

Strong Equivalence in PEPA

Definition

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

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

where

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

Weak equivalence relations

Weak equivalence relations (i.e. abstracting the internal τ actions) are difficult to obtain for stochastic process algebra which have integrated time and action, although there some results from Bernardo *et al*.

Weak equivalence relations

Weak equivalence relations (i.e. abstracting the internal τ actions) are difficult to obtain for stochastic process algebra which have integrated time and action, although there some results from Bernardo *et al*.

Further results on weak bisimulation have been obtained for SPA with orthogonal time and action such as IMC but even in this case there are some subtleties and it is not straightforward.

Weak equivalence relations

Weak equivalence relations (i.e. abstracting the internal τ actions) are difficult to obtain for stochastic process algebra which have integrated time and action, although there some results from Bernardo *et al*.

Further results on weak bisimulation have been obtained for SPA with orthogonal time and action such as IMC but even in this case there are some subtleties and it is not straightforward.

For PEPA the closest has been the definition of weak isomorphism which collapses a sequence of τ actions to a single τ action with the same mean duration.

Weak equivalence relations

Weak equivalence relations (i.e. abstracting the internal τ actions) are difficult to obtain for stochastic process algebra which have integrated time and action, although there some results from Bernardo *et al*.

Further results on weak bisimulation have been obtained for SPA with orthogonal time and action such as IMC but even in this case there are some subtleties and it is not straightforward.

For PEPA the closest has been the definition of weak isomorphism which collapses a sequence of τ actions to a single τ action with the same mean duration.

In general this will be an approximation of the original model from a dynamic perspective because the distribution of a sequence of exponential distributions is not an exponential distribution, although syntactic conditions have been identified for when this will be exact due to insensitivity.

Other equivalences

Whilst most work on equivalences in SPAs have focussed on bisimulation style equivalences, Marco Bernardo and co-authors have developed branching and testing equivalences in the context of the stochastic process algebra EMPA, and consequently for CTMCs.

Outline

A logical foundation for the specification language

The expression, and testing for satisfaction of equilibrium properties, can be seen to be closely related to the specification, and model checking of a formula expressed in Larsen and Skou's probabilistic modal logic (PML). We give a modified interpretation of such formulae suitable for reasoning about PEPA's continuous time models.

A logical foundation for the specification language

The expression, and testing for satisfaction of equilibrium properties, can be seen to be closely related to the specification, and model checking of a formula expressed in Larsen and Skou's probabilistic modal logic (PML). We give a modified interpretation of such formulae suitable for reasoning about PEPA's continuous time models.

We exploit the operators of modal logic to be more discriminating about which states contribute to the reward measure. In particular, we can select a state based on model behaviour which is not immediately local to the state.

Larsen and Skou's PML

$$
\begin{array}{lcll}
F & ::= & \mathsf{tt} & (\text{truth}) \\
 & | & \nabla\alpha & (\text{inability}) \\
 & | & \neg F & (\text{negation}) \\
 & | & F_1 \wedge F_2 & (\text{conjunction}) \\
 & | & \langle\alpha\rangle_\mu F & (\text{“at least”})
\end{array}
$$

Relation to PEPA

Defn. $P \xrightarrow{(\alpha,\nu)} S$ if for all $P' \in S$, $P \xrightarrow{\alpha} P'$ and $\sum\{r \mid P \xrightarrow{(\alpha,r)} P', P' \in S\} = \nu$.

Relation to PEPA

Defn. $P \xrightarrow{(\alpha,\nu)} S$ if for all $P' \in S$, $P \xrightarrow{\alpha} P'$ and $\sum\{r \mid P \xrightarrow{(\alpha,r)} P', P' \in S\} = \nu$.

Let P be a model of a PEPA process.

$$P \models \mathrm{tt}$$

Relation to PEPA

Defn. $P \xrightarrow{(\alpha,\nu)} S$ if for all $P' \in S$, $P \xrightarrow{\alpha} P'$ and $\sum\{r \mid P \xrightarrow{(\alpha,r)} P', P' \in S\} = \nu$.

Let P be a model of a PEPA process.

$$P \models \mathrm{tt}$$

$$P \models \neg F \text{ if } P \not\models F$$

Relation to PEPA

Defn. $P \xrightarrow{(\alpha,\nu)} S$ if for all $P' \in S$, $P \xrightarrow{\alpha} P'$ and $\sum\{r \mid P \xrightarrow{(\alpha,r)} P', P' \in S\} = \nu$.

Let P be a model of a PEPA process.

$$P \models \mathrm{tt}$$

$$P \models \neg F \text{ if } P \not\models F$$

$$P \models F_1 \wedge F_2 \text{ if } P \models F_1 \text{ and } P \models F_2$$

Relation to PEPA

Defn. $P \xrightarrow{(\alpha,\nu)} S$ if for all $P' \in S$, $P \xrightarrow{\alpha} P'$ and $\sum\{r \mid P \xrightarrow{(\alpha,r)} P', P' \in S\} = \nu$.

Let P be a model of a PEPA process.

$$P \models \mathrm{tt}$$

$$P \models \neg F \text{ if } P \not\models F$$

$$P \models F_1 \wedge F_2 \text{ if } P \models F_1 \text{ and } P \models F_2$$

$$P \models \nabla_\alpha \text{ if } P \xrightarrow{\alpha}\!\!\!\!\!\!/\;\;$$

Relation to PEPA

Defn. $P \xrightarrow{(\alpha,\nu)} S$ if for all $P' \in S$, $P \xrightarrow{\alpha} P'$ and $\sum\{r \mid P \xrightarrow{(\alpha,r)} P', P' \in S\} = \nu$.

Let P be a model of a PEPA process.

$$
\begin{aligned}
P &\models \mathrm{tt} \\
P &\models \neg F \text{ if } P \not\models F \\
P &\models F_1 \wedge F_2 \text{ if } P \models F_1 \text{ and } P \models F_2 \\
P &\models \nabla_\alpha \text{ if } P \xrightarrow{\alpha} \\
P &\models \langle\alpha\rangle_\mu F \text{ if } P \xrightarrow{(\alpha,\nu)} S \text{ for some } \nu \geq \mu, \\
&\qquad \text{and for all } P' \in S, P' \models F
\end{aligned}
$$

Modal characterisation of strong equivalence

Let P be a model of a PEPA process. Then

$$P \cong Q \text{ iff for all } F, P \models F \text{ iff } Q \models F$$

That is to say that two PEPA processes are strongly equivalent (in particular, their underlying Markov chains are lumpably equivalent) if and only if they both satisfy, in the setting where rates are quantified, the same set of PML formulae.