A Uniform Framework for Modeling Processes Behaviors and their Performances.

Rocco De Nicola

Dipartimento di Sistemi e Informatica
Università degli Studi di Firenze

ICE-PaCo@DISCOTEC 2011

Based on joint work with:
M. Bernardo, D. Latella, M. Loreti, M. Massink

Outline

Outline

Formal Methods for Reactive Systems

Our area of interest is reactive systems and the search of languages, methodologies and tools for guaranteeing their correct and efficient behavior in all possible environment.

To achieve this goal we need:

1. To study mathematical models for the formal description and analysis of concurrent programs.
2. To devise formal languages for the specification of the possible behaviour of parallel and reactive systems.
3. To develop verification tools and implementation techniques underlying them.

The basic approach

Processes and Process Description Languages

- The chosen abstraction for modelling reactive systems is the notion of process.
- Systems evolution is based on processes transformation: A process performs an action and becomes another process.
- Everything is (or can be viewed as) a process. Buffers, memory cells, tuple spaces, senders, receivers, . . . are all processes.
- Labelled Transition Systems (LTS) describe process behaviour (evolution from one stare to another), and permit directly modelling systems interaction.
- Languages are needed to describe concisely describe processes.

Presentations of Labelled Transition Systems

Process Description Languages as denotations of LTS

- LTS are represented by terms of a process description language, sometimes also referred as process algebra or process calculus.
- Terms of a process description language are rendered as LTS via operational semantics.

Process Algebra Basic Principles

1. Define a few elementary (atomic) processes modelling the simplest process behaviour;
2. Define appropriate composition operations to build more complex process behaviour from (existing) simpler ones.

Semantics of Process Calculi

Operational Semantics

An LTS is associated to each process term (built using the carefully selected set of operators) by relying on structural induction and on inference systems to define the meaning of each operator.

- the states of the transition systems are just terms of the Process Calculus (PC)
- the labels of the transitions connecting states represent the possible actions, or interactions, and their effects.

Behavioural Relations

PCs often come equipped with observational mechanisms that permit relating (through behavioral equivalences or preorders) systems according to their reactions to stimuli by external observers.

Semantics of Process Calculi

Definition (Inference Systems)

An inference system is a set of inference rules of the form

$$\frac{p_1, \cdots, p_n}{q}$$

For a generic operator *op* we have one or more rules like:

Inference Rules

$$\frac{E_{i_1} \xrightarrow{\alpha_1} E'_{i_1} \quad \cdots \quad E_{i_m} \xrightarrow{\alpha_m} E'_{i_m}}{op(E_1, \cdots, E_n) \xrightarrow{\alpha} op(E'_1, \cdots, E'_n)} \qquad \text{where } \{i_1, \cdots, i_m\} \subseteq \{1, \cdots, n\}.$$

From Behaviours to Performances

Functional Specifications

Initially, PCs have been designed for modeling *qualitative* aspects of concurrent systems:

- to model functional (extensional) behavior
- to assess whether two systems have comparable behaviors

Quantitative Specifications

However, it was soon noticed that other aspects of concurrent systems, mainly related to systems performance, actions duration and probability, are at least as important as the functional ones.

Introduction

Many variants of PCs have been introduced to take into account *quantitative* aspects of concurrent systems

- *deterministically timed* PCs;
- *probabilistic* PCs;
- *stochastically timed* PCs.

The operational semantics of these calculi has then been rendered in terms of richer LTSs quotiented with new (*timed*, *probabilistic* and *stochastic*) behavioral relations.

Introduction

Many variants of PCs have been introduced to take into account *quantitative* aspects of concurrent systems

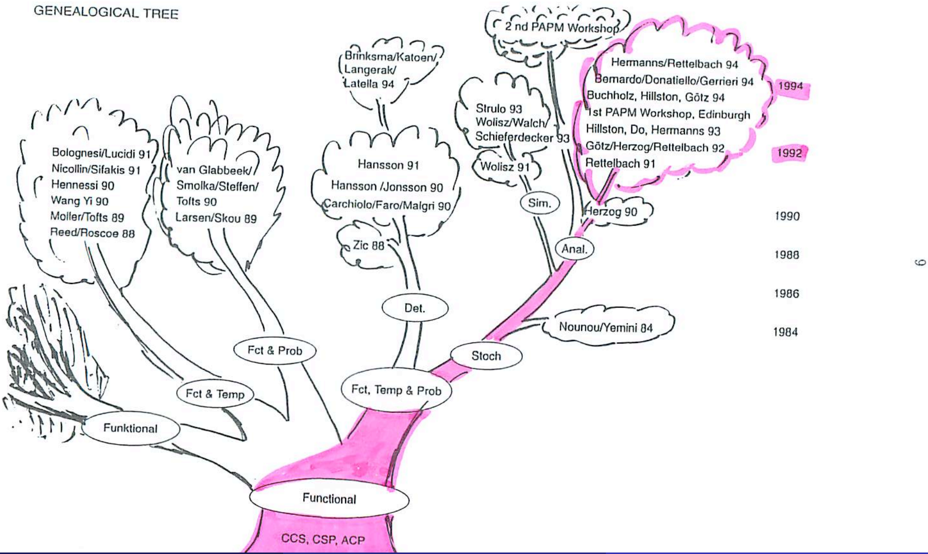
- *deterministically timed* PCs;
- *probabilistic* PCs;
- *stochastically timed* PCs.

The operational semantics of these calculi has then been rendered in terms of richer LTSs quotiented with new (*timed*, *probabilistic* and *stochastic*) behavioral relations.

Here, we will concentrate on some of these variants and will consider a few proposals for Stocastic Process Calculi - SPC and then we will touch some of the probabilistic variants.

Outline

Stochastic Process Calculi: a solid research field



Prominent Examples of SPCs

- **TIPP** - Timed Processes and Performance Evaluation *[N. Götz et al. 1993, Hermanns et al. 1998]*
- **PEPA** - Performance Evaluation Process Algebra *[Hillston 1996]*
- **EMPA** - Extended Markovian Process Algebra *[Bernardo et al. 1996]*
- **IML** - Language of Interactive Markov Chains *[Hermanns 2002]*
- **SπC** - Stochastic π-Calculus *[Priami, 1995]*
- **sCCS** - Stochastic CCS *[Klin & Sassone, 2008]*

Prominent Examples of SPCs

- **TIPP** - Timed Processes and Performance Evaluation *[N. Götz et al. 1993, Hermanns et al. 1998]*
- **PEPA** - Performance Evaluation Process Algebra *[Hillston 1996]*
- **EMPA** - Extended Markovian Process Algebra *[Bernardo et al. 1996]*
- **IML** - Language of Interactive Markov Chains *[Hermanns 2002]*

are based on CSP multi-party process interaction framework

- **S**π**C** - Stochastic π-Calculus *[Priami, 1995]*
- **sCCS** - Stochastic CCS *[Klin & Sassone, 2008]*

Prominent Examples of SPCs

- **TIPP** - Timed Processes and Performance Evaluation *[N. Götz et al. 1993, Hermanns et al. 1998]*
- **PEPA** - Performance Evaluation Process Algebra *[Hillston 1996]*
- **EMPA** - Extended Markovian Process Algebra *[Bernardo et al. 1996]*
- **IML** - Language of Interactive Markov Chains *[Hermanns 2002]*

are based on CSP multi-party process interaction framework

- **SπC** - Stochastic π-Calculus *[Priami, 1995]*
- **sCCS** - Stochastic CCS *[Klin & Sassone, 2008]*

are based on CCS binary process interaction framework

SPC for Mobile / Service Oriented Computing

- **StoKLAIM** - Stochastic KLAIM *[De Nicola et al. 2005-9]*
  Kernel Language for Agents Interaction and Mobility *[De Nicola et al. 1998]*
- **MarCaSPiS** - Markovian CaSPiS (RTS semantics) *[De Nicola et al. 2008]*
  Calculus of Sessions and Pipelines *[Bruni et al. 2008]*
- **Stochastic COWS** *[Prandi et al. 2007]*
  Calculus for Orchestration of Web Services *[Lapadula et al. 2007]*

Stochastically Timed Process Calculi

Goals:

- Integration of
  - qualitative (behavioural, functional) system model descriptions with
  - quantitative (non-functional, e.g. performance/dependability) ones

  in a single mathematical process algebraic framework with
  - Formal Syntax
  - Process Semantic Models

    Pre-orders, Equivalence relations, Axiomatizations, etc.
  - Formal Analysis and Verification

    Traditional Techniques, (Stochastic) Logics & Model-checking, etc.

Means:

- Enriching process languages with random variables (RV) modeling
  - action durations or delays before instantaneous actions
- Combining
  - Labeled Transition Systems (LTS) with
  - Continuous Time Markov Chains (CTMC)

Continuous Time Markov Chains

Continuous Time Markov Chains are a successful mathematical framework for modeling and analysing performance and dependability of systems that rely on exponential distribution of states transitions.

CTMCs come with

- Well established Analysis Techniques
  - Steady State Analysis
  - Transient Analysis
- Efficient Software Tools based on:
  - Stochastic Timed/Temporal Logics
  - Stochastic Model Checking

Continuous Time Markov Chains

Continuous Time Markov Chains are a successful mathematical framework for modeling and analysing performance and dependability of systems that rely on exponential distribution of states transitions.

CTMCs come with

- Well established Analysis Techniques
  - Steady State Analysis
  - Transient Analysis
- Efficient Software Tools based on:
  - Stochastic Timed/Temporal Logics
  - Stochastic Model Checking

A CTMC is a pair $(\mathcal{S}, \mathbf{R})$

- $\mathcal{S}$: a countable set of states
- $\mathbf{R} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$, the rate matrix

Stochastic process calculi

- A CTMC is associated to each process term;
- CTMCs model the stochastic behaviour of processes.

Stochastic process calculi

- A CTMC is associated to each process term;
- CTMCs model the stochastic behaviour of processes.

To get a CTMC from a term, one needs to...

- compute *synchronizations rate* ...

Stochastic process calculi

- A CTMC is associated to each process term;
- CTMCs model the stochastic behaviour of processes.

To get a CTMC from a term, one needs to...

- compute *synchronizations rate* ...
- ...while taking into account transition multiplicity, for determining correct execution rate

Stochastic process calculi

- A CTMC is associated to each process term;
- CTMCs model the stochastic behaviour of processes.

To get a CTMC from a term, one needs to...

- compute *synchronizations rate* ...
- ...while taking into account transition multiplicity, for determining correct execution rate

Process Calculi:

$$\alpha.P + \alpha.P = \alpha.P$$

$$recX\ \alpha.\ X \mid recX\ \alpha.\ X = recX\ \alpha.\ X$$

Stochastic process calculi

- A CTMC is associated to each process term;
- CTMCs model the stochastic behaviour of processes.

To get a CTMC from a term, one needs to. . .

- compute *synchronizations rate* . . .
- . . . while taking into account transition multiplicity, for determining correct execution rate

Stochastic Process Calculi:

$$\alpha^{\lambda}.P + \alpha^{\lambda}.P \quad \neq \quad \alpha^{\lambda}.P$$

$$recX\ \alpha^{\lambda}.\ X \mid recX\ \alpha^{\lambda}.\ X \quad \neq \quad recX\ \alpha^{\lambda}.\ X$$

Stochastic process calculi

- A CTMC is associated to each process term;
- CTMCs model the stochastic behaviour of processes.

To get a CTMC from a term, one needs to...

- compute *synchronizations rate* ...
- ...while taking into account transition multiplicity, for determining correct execution rate

Stochastic Process Calculi:

$$\alpha^{\lambda}.P + \alpha^{\lambda}.P = \alpha^{2\lambda}.P$$

$$recX\ \alpha^{\lambda}.\ X \mid recX\ \alpha^{\lambda}.\ X = recX\ \alpha^{2\lambda}.\ X$$

SPCs: Similarities & Differences

Most of SPA are based on the same Exponentially distributed RV, fully characterized by their rate, but they differ significantly for

- the process interaction paradigm (e.g CSP-like vs CCS-like)

SPCs: Similarities & Differences

Most of SPA are based on the same Exponentially distributed RV, fully characterized by their rate, but they differ significantly for

- the process interaction paradigm (e.g CSP-like vs CCS-like)
- the association of rates with actions
  - *rate-action-prefix*: $(a, \lambda).P$ *[e.g. TIPP, PEPA, EMPA, sCCS, SπC]*
  - *rate-prefix* plus *action-prefix*: $\lambda.P$, $a.P$ *[e.g. IML]*

SPCs: Similarities & Differences

Most of SPA are based on the same Exponentially distributed RV, fully characterized by their rate, but they differ significantly for

- the process interaction paradigm (e.g CSP-like vs CCS-like)
- the association of rates with actions
  - *rate-action-prefix*: $(a, \lambda).P$ *[e.g. TIPP, PEPA, EMPA, sCCS, SπC]*
  - *rate-prefix* plus *action-prefix*: $\lambda.P$, $a.P$ *[e.g. IML]*
- the definition of the rate associated to synchronisations

SPCs: Similarities & Differences

Most of SPA are based on the same Exponentially distributed RV, fully characterized by their rate, but they differ significantly for

- the process interaction paradigm (e.g CSP-like vs CCS-like)
- the association of rates with actions
  - *rate-action-prefix*: $(a, \lambda).P$ *[e.g. TIPP, PEPA, EMPA, sCCS, SπC]*
  - *rate-prefix* plus *action-prefix*: $\lambda.P$, $a.P$ *[e.g. IML]*
- the definition of the rate associated to synchronisations
- the modelling of the choice between equal behaviours
  - multi-relations *[e.g. PEPA, IML]*
  - proved transition systems *[e.g. TIPP, SπC]*
  - LTS with numbered transitions *[e.g. LCTMC]*
  - unique rate names *[e.g. StoKLAIM]*

Operators for Stochastic Process Calculi

A uniform syntax for many SPCs

P, Q	$::=$	**nil**	[*inaction*]
	$\mid$	$\lambda.P$	[*rate prefix*]
	$\mid$	$a.P$	[*action prefix*]
	$\mid$	$\langle a, \lambda \rangle.P$	[*rated-action prefix*]
	$\mid$	$\langle a, *_\omega \rangle.P$	[*passive-action prefix*]
	$\mid$	$\bar{a}^\lambda.P$	[*rated-output-action prefix*]
	$\mid$	$a^\lambda.P$	[*rated-input-action prefix*]
	$\mid$	$a^{*_\omega}.P$	[*passive-input-action prefix*]
	$\mid$	$P + Q$	[*choice composition*]
	$\mid$	$P \;\|_L\; Q$	[*multi-party synchronization composition*]
	$\mid$	$P \mid Q$	[*binary synchronization composition*]
	$\mid$	X	[*constant*]

Operators for Stochastic Process Calculi

- Rate prefix: $\lambda.P$, delays execution of P by an interval δ^e, the duration of which is an exponentially distributed RV with rate $\lambda \in \mathbb{R}_{>0}$.

Operators for Stochastic Process Calculi

- Rate prefix: $\lambda.P$, delays execution of P by an interval δ^e, the duration of which is an exponentially distributed RV with rate $\lambda \in \mathbb{R}_{>0}$.
- Action prefix: $a.P$ starts with the execution of action a and then continues with that of P; the execution of a is *duration-less* or instantaneous, i.e. takes no time.

Operators for Stochastic Process Calculi

- Rate prefix: $\lambda.P$, delays execution of P by an interval δ^e, the duration of which is an exponentially distributed RV with rate $\lambda \in \mathbb{R}_{>0}$.
- Action prefix: $a.P$ starts with the execution of action a and then continues with that of P; the execution of a is *duration-less* or instantaneous, i.e. takes no time.
- In rated-action prefix: $\langle a, \lambda \rangle.P$ the duration δ^e_a of the execution of action a is an exponentially distributed RV with rate λ; after completion of the execution of a, the behavior continues as in P.

Operators for Stochastic Process Calculi

- Rate prefix: $\lambda.P$, delays execution of P by an interval δ^e, the duration of which is an exponentially distributed RV with rate $\lambda \in \mathbb{R}_{>0}$.
- Action prefix: $a.P$ starts with the execution of action a and then continues with that of P; the execution of a is *duration-less* or instantaneous, i.e. takes no time.
- In rated-action prefix: $\langle a, \lambda \rangle.P$ the duration δ^e_a of the execution of action a is an exponentially distributed RV with rate λ; after completion of the execution of a, the behavior continues as in P.
- The weight ω in passive-action prefix: operator $\langle a, *_\omega \rangle.P$ is used for determining a probabilistic distribution in case there is more than one passive action which may synchronize with the same active one.

Operators for Stochastic Process Calculi

- Rated-input-action prefix: $a^{\lambda}.P$, and rated-output-action prefix: $\bar{a}^{\lambda}.P$ are used to model CCS-like stochastic calculi, where a *binary* synchronization paradigm is used. In this calculi duration rates are associated to both to input and output actions.

Operators for Stochastic Process Calculi

- Rated-input-action prefix: $a^\lambda.P$, and rated-output-action prefix: $\bar{a}^\lambda.P$ are used to model CCS-like stochastic calculi, where a *binary* synchronization paradigm is used. In this calculi duration rates are associated to both to input and output actions.
- The choice operator: $P_1 + P_2$ is interpreted according to the *race condition* principle of CTMCs.

Operators for Stochastic Process Calculi

- Rated-input-action prefix: $a^{\lambda}.P$, and rated-output-action prefix: $\bar{a}^{\lambda}.P$ are used to model CCS-like stochastic calculi, where a *binary* synchronization paradigm is used. In this calculi duration rates are associated to both to input and output actions.
- The choice operator: $P_1 + P_2$ is interpreted according to the *race condition* principle of CTMCs.
- The multi-party parallel composition operator: $P_1 \; ||_L \; P_2$ where $L \in (\wp_{fin}\,\mathcal{A})$ is the synchronization (or cooperation) set, corresponds to CSP parallel composition that requires actions in L to be performed synchronously and the others independently.

Operators for Stochastic Process Calculi

- Rated-input-action prefix: $a^{\lambda}.P$, and rated-output-action prefix: $\bar{a}^{\lambda}.P$ are used to model CCS-like stochastic calculi, where a *binary* synchronization paradigm is used. In this calculi duration rates are associated to both to input and output actions.
- The choice operator: $P_1 + P_2$ is interpreted according to the *race condition* principle of CTMCs.
- The multi-party parallel composition operator: $P_1 \parallel_L P_2$ where $L \in (\wp_{fin}\,\mathcal{A})$ is the synchronization (or cooperation) set, corresponds to CSP parallel composition that requires actions in L to be performed synchronously and the others independently.
- The binary parallel composition: $P_1 \mid P_2$, is the parallel operator used in the CCS-based calculi, that models synchronization of complementary actions. For this composition also passive input action prefix: $a^{*\omega}.P$ is used.

SPCs: Headaches

Transition multiplicity (race condition)

- The technicalities set up for dealing with transition multiplicity often blur the conceptual understanding of the calculus.
- The transition multi-relation defined as the *least multi-relation* induced by a set of SOS rules (unintentionally!) boils down to a *relation*.

Interaction paradigm and synchronisation rate

- Use of classical SOS for CCS-like interaction in combination with the *minimal apparent rate* principle may lead to *loss of associativity* for the parallel composition operator.

Outline

Semantics of stochastic process calculi

We introduce a variant of Transition Systems (that we call RTS) and use it for defining stochastic behaviour of a few process algebras. Our RTS associates terms and actions to functions from terms to rates

Semantics of stochastic process calculi

We introduce a variant of Transition Systems (that we call RTS) and use it for defining stochastic behaviour of a few process algebras. Our RTS associates terms and actions to functions from terms to rates

Like most of the previous attempts we take a two steps approach: For a given term, say T, we define an enriched LTS and then use it to determine the CTMC to be associated to T.

Semantics of stochastic process calculi

Stochastic semantics of process calculi is defined by means of a transition relation $\rightarrowtail$ that associates to a pair (P, α) - consisting of process and an action - a total function $(\mathscr{P}, \mathscr{Q}, \ldots)$ that assigns a non-negative real number to each process of the calculus. Value 0 is assigned to unreachable processes.

Semantics of stochastic process calculi

Stochastic semantics of process calculi is defined by means of a transition relation $\rightarrowtail$ that associates to a pair (P, α) - consisting of process and an action - a total function $(\mathscr{P}, \mathscr{Q}, \ldots)$ that assigns a non-negative real number to each process of the calculus. Value 0 is assigned to unreachable processes.

$P \overset{\alpha}{\rightarrowtail} \mathscr{P}$ means that, for a generic process Q:

- if $\mathscr{P}(Q) = x$ $(\neq 0)$ then Q is reachable from P via the execution of α with rate/(weight) x
- if $\mathscr{P}(Q) = 0$ then Q is not reachable from P via α

Semantics of stochastic process calculi

Stochastic semantics of process calculi is defined by means of a transition relation $\rightarrowtail$ that associates to a pair (P, α) - consisting of process and an action - a total function $(\mathscr{P}, \mathscr{Q}, \ldots)$ that assigns a non-negative real number to each process of the calculus. Value 0 is assigned to unreachable processes.

$P \xrightarrow{\alpha} \mathscr{P}$ means that, for a generic process Q:

- if $\mathscr{P}(Q) = x$ $(\neq 0)$ then Q is reachable from P via the execution of α with rate/(weight) x
- if $\mathscr{P}(Q) = 0$ then Q is not reachable from P via α

We have that if $P \xrightarrow{\alpha} \mathscr{P}$ then

- $\oplus\mathscr{P} = \sum_Q \mathscr{P}(Q)$ represents the total rate/weight of α in P.

Rate transition systems

Definition

A rate transition system is a triple $(S, A, \rightarrowtail)$ where:

- S is a set of states;
- A is a set of transition labels;
- $\rightarrowtail \subseteq S \times A \times [S \rightarrow \mathbb{R}_{\geq 0}]$

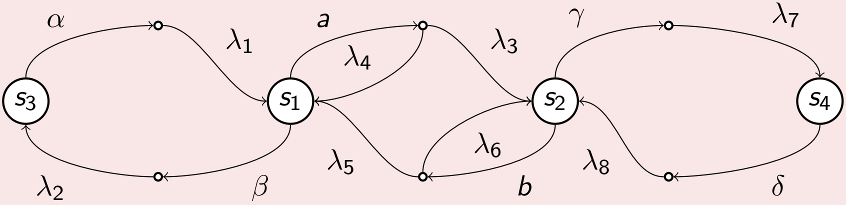
Rate transition systems

Definition

A rate transition system is a triple $(S, A, \rightarrowtail)$ where:

- S is a set of states;
- A is a set of transition labels;
- $\rightarrowtail \subseteq S \times A \times [S \rightarrow \mathbb{R}_{\geq 0}]$

An example of RTS



Some Notation for Rate transition systems

- RTS will be denoted by $\mathcal{R}$, $\mathcal{R}_1$, $\mathcal{R}'$, ...,
- Elements of $[S \to \mathbb{R}_{\geq 0}]$ are denoted by $\mathscr{P}, \mathscr{Q}, \mathscr{R}, \ldots$
- $[s_1 \mapsto v_1, \ldots, s_n \mapsto v_n]$ denotes the function associating v_i to s_i and 0 to all the other states.
- $[]$ denotes the constant function 0.
- χ_s stands for $[s \mapsto 1]$.
- $\mathscr{P} + \mathscr{Q}$ denotes the function $\mathscr{R}$ such that: $\mathscr{R}(s) = \mathscr{P}(s) + \mathscr{Q}(s)$.
- $\mathscr{P} \cdot \frac{x}{y}$ denotes the function $\mathscr{R}$ such that: $\mathscr{R}(s) = \mathscr{P}(s) \cdot \frac{x}{y}$ if $y \neq 0$, and $\emptyset$ if $y = 0$.

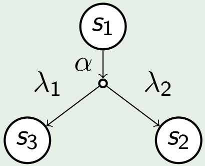
Rate transition systems

Definition

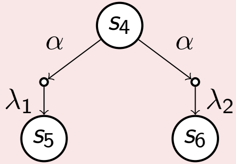
Let $\mathcal{R} = (S, A, \rightarrow)$ be an RTS

- $\mathcal{R}$ is *total*: $\forall s \in S$, $\forall \alpha \in A$ $\exists \mathscr{P}$ such that $s \xrightarrow{\alpha} \mathscr{P}$;
- $\mathcal{R}$ is *deterministic*: $\forall s \in S$, $\forall \alpha \in A$ $s \xrightarrow{\alpha} \mathscr{P}$, $s \xrightarrow{\alpha} \mathscr{Q} \Longrightarrow \mathscr{P} = \mathscr{Q}$
- $\mathcal{R}$ is a *finite support*: $\forall s \in S$, $\forall \alpha \in A$ if $s \xrightarrow{\alpha} \mathscr{P}$ we then $\{s' | \mathscr{P}(s') > 0\}$ is finite

A deterministic RTS



A general RTS



From RTS to CTMC...

Reachable Sets of States

For sets $\mathcal{S}' \subseteq \mathcal{S}$ and $A' \subseteq A$, the set of derivatives of $\mathcal{S}'$ *through* A', denoted $Der(\mathcal{S}', A')$, is the smallest set such that:

- $\mathcal{S}' \subseteq Der(\mathcal{S}', A')$,
- if $s \in Der(\mathcal{S}', A')$ and there exists $\alpha \in A'$ and $\mathscr{Q} \in \Sigma_{\mathcal{S}}$ such that $s \stackrel{\alpha}{\rightarrowtail} \mathscr{Q}$ then $\{s' \mid \mathscr{Q}(s') > 0\} \subseteq Der(\mathcal{S}', A')$

Mapping $(\mathcal{S}, A, \rightarrowtail)$ into $(Der(\mathcal{S}', A'), \mathbf{R})$

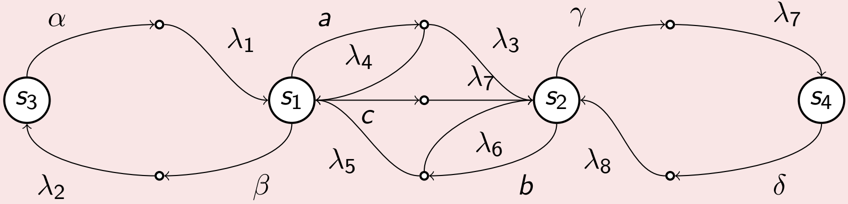
Let $\mathcal{R} = (\mathcal{S}, A, \rightarrowtail)$ be a *functional* RTS, for $\mathcal{S}' \subseteq \mathcal{S}$, the CTMC of $\mathcal{S}'$, when one considers only actions $A' \subseteq A$ is defined as $CTMC[\mathcal{S}', A'] =_{\mathrm{def}} (Der(\mathcal{S}', A'), \mathbf{R})$ where for all $s_1, s_2 \in Der(\mathcal{S}', A')$:

$$\mathbf{R}[s_1, s_2] =_{\mathrm{def}} \sum_{\alpha \in A'} \mathscr{P}^{\alpha}(s_2) \qquad \text{with } s_1 \stackrel{\alpha}{\rightarrowtail} \mathscr{P}^{\alpha}.$$

A translation from an RTS to a CTMC

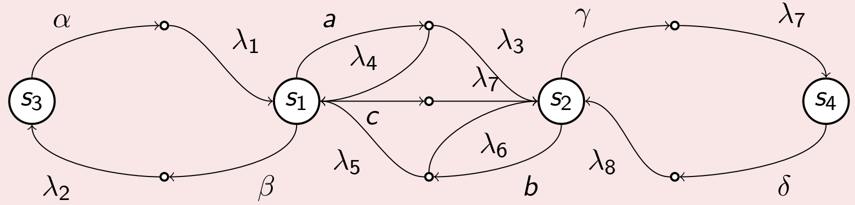
A translation from an RTS to a CTMC

An RTS:

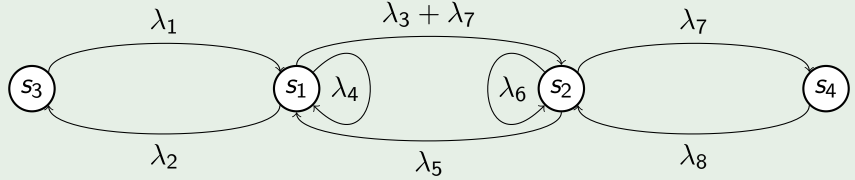


A translation from an RTS to a CTMC

An RTS:

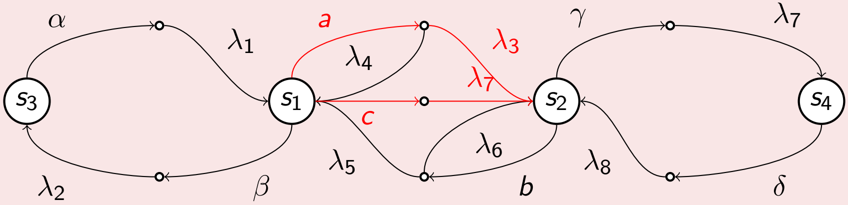


The corresponding CTMC:

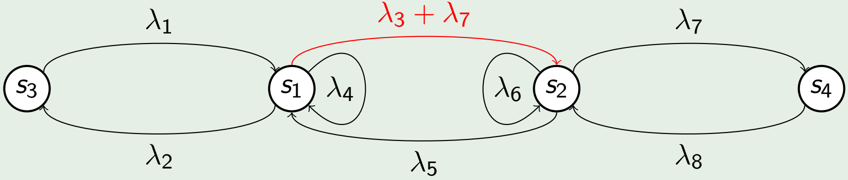


A translation from an RTS to a CTMC

An RTS:



The corresponding CTMC:



Outline

Outline

A Language for CTMCs - Hermanns et al. 2002

Syntax

$$
\begin{array}{llll}
P, Q & ::= & \mathbf{nil} & [\mathit{inaction}] \\
 & \mid & \lambda.P & [\mathit{rate\ prefix}] \\
 & \mid & P + P & [\mathit{choice\ composition}] \\
 & \mid & X & [\mathit{constant}]
\end{array}
$$

with $\lambda \in \mathbb{R}_{>0}$ and with definition $X := P$ such that all process constants are *guarded* in P.

A Language for CTMCs - Hermanns et al. 2002

Syntax

$$
\begin{array}{llll}
P, Q & ::= & \mathbf{nil} & [\textit{inaction}] \\
 & \mid & \lambda.P & [\textit{rate prefix}] \\
 & \mid & P + P & [\textit{choice composition}] \\
 & \mid & X & [\textit{constant}]
\end{array}
$$

with $\lambda \in \mathbb{R}_{>0}$ and with definition $X := P$ such that all process constants are *guarded* in P.

Semantics Rules

Label set is $\mathcal{L}_{CTMC} =_{\text{def}} \{\delta^e\}$

$$
\frac{}{\mathbf{nil} \overset{\delta^e}{\rightarrowtail} []_{\mathbb{R}}} \qquad
\frac{}{\lambda.P \overset{\delta^e}{\rightarrowtail} [P \mapsto \lambda]} \qquad
\frac{P \overset{\delta^e}{\rightarrowtail} \mathscr{P}, Q \overset{\delta^e}{\rightarrowtail} \mathscr{Q}}{P + Q \overset{\delta^e}{\rightarrowtail} \mathscr{P} + \mathscr{Q}} \qquad
\frac{P \overset{\delta^e}{\rightarrowtail} \mathscr{P}, X := P}{X \overset{\delta^e}{\rightarrowtail} \mathscr{P}}
$$

Table 1: Transition Rules for the Language for CTMCs

Choice and Transition Multiplicity

Choice and Transition Multiplicity

$$\frac{P \overset{\delta^e}{\rightarrowtail} \mathscr{P},\ Q \overset{\delta^e}{\rightarrowtail} \mathscr{Q}}{P + Q \overset{\delta^e}{\rightarrowtail} \mathscr{P} + \mathscr{Q}}$$

Choice and Transition Multiplicity

$$\frac{P \overset{\delta^e}{\rightarrowtail} \mathscr{P},\ Q \overset{\delta^e}{\rightarrowtail} \mathscr{Q}}{P + Q \overset{\delta^e}{\rightarrowtail} \mathscr{P} + \mathscr{Q}}$$

Take $\lambda.R_1 + \mu.R_2$

Choice and Transition Multiplicity

$$\frac{P \overset{\delta^e}{\rightarrowtail} \mathscr{P},\ Q \overset{\delta^e}{\rightarrowtail} \mathscr{Q}}{P + Q \overset{\delta^e}{\rightarrowtail} \mathscr{P} + \mathscr{Q}}$$

Take $\lambda.R_1 + \mu.R_2$

- if $R_1 \neq R_2$:

Choice and Transition Multiplicity

$$\frac{P \stackrel{\delta^e}{\rightarrowtail} \mathscr{P},\ Q \stackrel{\delta^e}{\rightarrowtail} \mathscr{Q}}{P + Q \stackrel{\delta^e}{\rightarrowtail} \mathscr{P} + \mathscr{Q}}$$

Take $\lambda.R_1 + \mu.R_2$

- if $R_1 \neq R_2$: then $\lambda.R_1 + \mu.R_2 \stackrel{\delta^e}{\rightarrowtail} [R_1 \mapsto \lambda, R_2 \mapsto \mu]$

Choice and Transition Multiplicity

$$\frac{P \overset{\delta^e}{\rightarrowtail} \mathscr{P},\ Q \overset{\delta^e}{\rightarrowtail} \mathscr{Q}}{P + Q \overset{\delta^e}{\rightarrowtail} \mathscr{P} + \mathscr{Q}}$$

Take $\lambda.R_1 + \mu.R_2$

- if $R_1 \neq R_2$: then $\lambda.R_1 + \mu.R_2 \overset{\delta^e}{\rightarrowtail} [R_1 \mapsto \lambda, R_2 \mapsto \mu]$
- if $R_1 = R_2 = R$ then $\lambda.R + \mu.R \overset{\delta^e}{\rightarrowtail} [R \mapsto \lambda + \mu]$

  thus, we obviously have

Choice and Transition Multiplicity

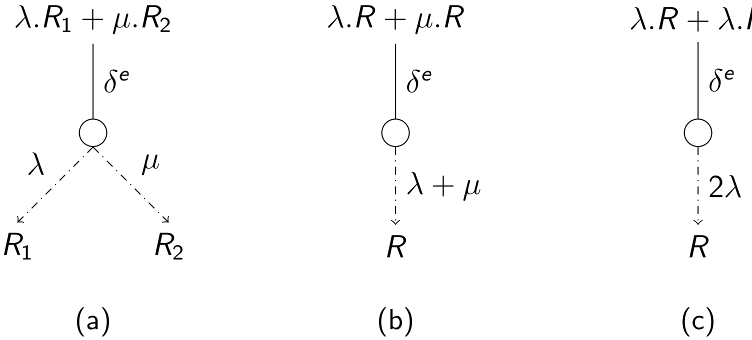
$$\frac{P \overset{\delta^e}{\rightarrowtail} \mathscr{P},\ Q \overset{\delta^e}{\rightarrowtail} \mathscr{Q}}{P + Q \overset{\delta^e}{\rightarrowtail} \mathscr{P} + \mathscr{Q}}$$

Take $\lambda.R_1 + \mu.R_2$

- if $R_1 \neq R_2$: then $\lambda.R_1 + \mu.R_2 \overset{\delta^e}{\rightarrowtail} [R_1 \mapsto \lambda, R_2 \mapsto \mu]$
- if $R_1 = R_2 = R$ then $\lambda.R + \mu.R \overset{\delta^e}{\rightarrowtail} [R \mapsto \lambda + \mu]$

  thus, we obviously have
- if $R_1 = R_2 = R$ and $\lambda = \mu$ then $\lambda.R + \lambda.R \overset{\delta^e}{\rightarrowtail} [R \mapsto 2 \cdot \lambda]$

Choice and Transition Multiplicity, pictorially



Handling parallel composition

Handling parallel composition

Let us consider a generic process language $\mathcal{P}_C$ providing a process parallel composition operator, denoted by, say, $\times$:

- reachable states of $P_1 \times P_2$ are obtained via a suitable composition of P_1, P_2, the states reachable from P_1, and the states reachable from P_2.

Handling parallel composition

Let us consider a generic process language $\mathcal{P}_C$ providing a process parallel composition operator, denoted by, say, $\times$:

- reachable states of $P_1 \times P_2$ are obtained via a suitable composition of P_1, P_2, the states reachable from P_1, and the states reachable from P_2.

If $\times$ is the *interleaving* operator then the continuation functions of $P_1 \times P_2$ on α-labelled transitions are obtained by composing

- the α-continuations of P_1 in parallel with P_2.
- P_1 in parallel with the α-continuations of P_2.

Handling parallel composition

Let us consider a generic process language $\mathcal{P}_C$ providing a process parallel composition operator, denoted by, say, $\times$:

- reachable states of $P_1 \times P_2$ are obtained via a suitable composition of P_1, P_2, the states reachable from P_1, and the states reachable from P_2.

If $\times$ is the *interleaving* operator then the continuation functions of $P_1 \times P_2$ on α-labelled transitions are obtained by composing

- the α-continuations of P_1 in parallel with P_2.
- P_1 in parallel with the α-continuations of P_2.

To provide a uniform description of the stochastic semantics of parallel composition, we have introduced a set of basic operators that can be composed to capture the semantics of the operators of each SPC.

Handling parallel composition

Parallel aggregation: $\mathscr{P} \otimes_{\times} \mathscr{Q}$

$$(\mathscr{P} \otimes_{\times} \mathscr{Q})\, s =_{\mathrm{def}} \begin{cases} (\mathscr{P}\, s_1) \cdot (\mathscr{Q}\, s_2), & \text{if } \exists s_1, s_2 \in S.\ s = s_1 \times s_2 \\ [], & \text{otherwise} \end{cases}$$

Handling parallel composition

Parallel aggregation: $\mathscr{P} \otimes_{\times} \mathscr{Q}$

$$(\mathscr{P} \otimes_{\times} \mathscr{Q})\, s =_{\mathrm{def}} \begin{cases} (\mathscr{P}\, s_1) \cdot (\mathscr{Q}\, s_2), & \text{if } \exists s_1, s_2 \in S.\ s = s_1 \times s_2 \\ [], & \text{otherwise} \end{cases}$$

Renormalization: $\mathscr{P} \cdot \frac{x}{y}$

$$\left(\mathscr{P} \cdot \frac{x}{y}\right) s =_{\mathrm{def}} \begin{cases} (\mathscr{P}\, s) \cdot (x/y), & \text{if } y \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

Handling parallel composition

Parallel aggregation: $\mathscr{P} \otimes_\times \mathscr{Q}$

$$(\mathscr{P} \otimes_\times \mathscr{Q})\, s =_{\mathrm{def}} \begin{cases} (\mathscr{P}\, s_1) \cdot (\mathscr{Q}\, s_2), & \text{if } \exists s_1, s_2 \in S.\ s = s_1 \times s_2 \\ [], & \text{otherwise} \end{cases}$$

Renormalization: $\mathscr{P} \cdot \frac{x}{y}$

$$\left(\mathscr{P} \cdot \frac{x}{y}\right) s =_{\mathrm{def}} \begin{cases} (\mathscr{P}\, s) \cdot (x/y), & \text{if } y \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

Characteristic functions: $(\mathcal{X}\ s)$

$$\mathcal{X}\ s = [s \mapsto 1]$$

Parallel Composition of CTMCs

Parallel Composition of CTMCs

To show how the operators introduced can be used, we extend the language of CTMC with the parallel operator $||$, where $P_1 \; || \; P_2$ identifies the *interleaving* between P_1 and P_2.

Parallel Composition of CTMCs

To show how the operators introduced can be used, we extend the language of CTMC with the parallel operator $||$, where $P_1 \ || \ P_2$ identifies the *interleaving* between P_1 and P_2.

In $P_1 \ || \ P_2$, P_1 and P_2 do not cooperate and the reachable states are those reachable from P_1 (respectively, P_2) composed in parallel with P_2 (respectively P_1).

Parallel Composition of CTMCs

To show how the operators introduced can be used, we extend the language of CTMC with the parallel operator $||$, where $P_1 \ || \ P_2$ identifies the *interleaving* between P_1 and P_2.

In $P_1 \ || \ P_2$, P_1 and P_2 do not cooperate and the reachable states are those reachable from P_1 (respectively, P_2) composed in parallel with P_2 (respectively P_1).

If $P_1 \xrightarrow{\delta^e} \mathscr{P}$ and $P_2 \xrightarrow{\delta^e} \mathscr{Q}$, the states reachable from $P_1 \ || \ P_2$ are obtained by *combining* $\mathscr{P}$ and $\mathscr{Q}$ respectively with P_2 and P_1:

- $\mathscr{P} \otimes_{||} (\mathcal{X}_{\mathbb{R}_{\geq 0}} P_2)$
  - the states reachable from P_1 in parallel with P_2
- $(\mathcal{X}_{\mathbb{R}_{\geq 0}} P_1) \otimes_{||} \mathscr{Q}$
  - P_1 in parallel with the states reachable from P_2.

Parallel Composition of CTMCs

The rule governing behaviour of parallel composed processes is the following:

$$\frac{P_1 \overset{\delta^e}{\rightarrowtail} \mathscr{P} \quad P_2 \overset{\delta^e}{\rightarrowtail} \mathscr{Q}}{P_1 \mid\mid P_2 \overset{\delta^e}{\rightarrowtail} (\mathscr{P} \otimes_{||} (\mathcal{X}_{\mathbb{R}}\, P_2)) + ((\mathcal{X}_{\mathbb{R}}\, P_1) \otimes_{||} \mathscr{Q})}$$

Parallel Composition of CTMCs

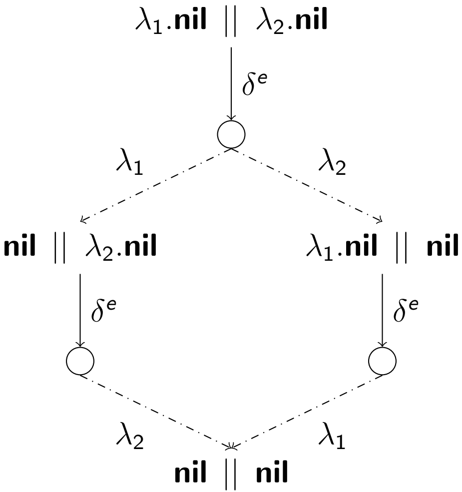
An example: $\lambda_1.\mathbf{nil} \mid\mid \lambda_2.\mathbf{nil}$

$$\frac{\lambda_1.\mathbf{nil} \overset{\delta^e}{\rightarrowtail} [\mathbf{nil} \mapsto \lambda_1] \qquad \lambda_2.\mathbf{nil} \overset{\delta^e}{\rightarrowtail} [\mathbf{nil} \mapsto \lambda_2]}{\lambda_1.\mathbf{nil} \mid\mid \lambda_2.\mathbf{nil} \;\overset{\delta^e}{\rightarrowtail}\; \begin{array}{l} [\mathbf{nil} \mapsto \lambda_1] \otimes_{||} (\mathcal{X}_{\mathbb{R}}\lambda_2.\mathbf{nil}) \\ +_{\mathbb{R}} \\ (\mathcal{X}_{\mathbb{R}}\lambda_1.\mathbf{nil}) \otimes_{||} [\mathbf{nil} \mapsto \lambda_2] \end{array}}$$

where:

$$\begin{array}{rl} & [\mathbf{nil} \mapsto \lambda_1] \otimes_{||} (\mathcal{X}_{\mathbb{R}}\lambda_2.\mathbf{nil}) +_{\mathbb{R}} (\mathcal{X}_{\mathbb{R}}\lambda_1.\mathbf{nil}) \otimes_{||} [\mathbf{nil} \mapsto \lambda_2] \\ = & [\mathbf{nil} \mapsto \lambda_1] \otimes_{||} [\lambda_2.\mathbf{nil} \mapsto 1_{\mathbb{R}}] +_{\mathbb{R}} [\lambda_1.\mathbf{nil} \mapsto 1_{\mathbb{R}}] \otimes_{||} [\mathbf{nil} \mapsto \lambda_2] \\ = & [\mathbf{nil} \mid\mid \lambda_2.\mathbf{nil} \mapsto \lambda_1] +_{\mathbb{R}} [\lambda_1.\mathbf{nil} \mid\mid \mathbf{nil} \mapsto \lambda_2] \\ = & [\mathbf{nil} \mid\mid \lambda_2.\mathbf{nil} \mapsto \lambda_1 \;,\; \lambda_1.\mathbf{nil} \mid\mid \mathbf{nil} \mapsto \lambda_2] \end{array}$$

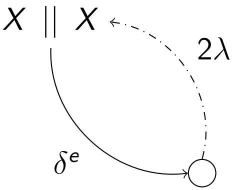
Parallel Composition of CTMCs



Parallel Composition of CTMCs

Example: $X \mid\mid X$ where $X := \lambda.X$

$$\frac{X \overset{\delta^e}{\rightarrowtail} [X \mapsto \lambda] \qquad X \overset{\delta^e}{\rightarrowtail} [X \mapsto \lambda]}{X \mid\mid X \overset{\delta^e}{\rightarrowtail} [X \mapsto \lambda] \otimes_{\mid\mid} (\mathcal{X}_{\mathbb{R}} X) +_{\mathbb{R}} (\mathcal{X}_{\mathbb{R}} X) \otimes_{\mid\mid} [X \mapsto \lambda]}$$



Outline

TIPP - Hermanns et al. 2002

TIPP Operators

- *inaction*: **nil**
- *rated-action prefix*: $\langle a, \lambda \rangle . P$
- *choice*: $P + Q$
- *multi-party synchronization* $P \; ||_L \; Q$
- *constant*: X (where $X := P$)

TIPP - Hermanns et al. 2002

TIPP Operators

- *inaction*: **nil**
- *rated-action prefix*: $\langle a, \lambda \rangle .P$
- *choice*: $P + Q$
- *multi-party synchronization* $P \;||_L\; Q$
- *constant*: X (where $X := P$)

The rate of a synchronization is obtained as the product of the rates of involved actiions.

TIPP - Hermanns et al. 2002

Operational Semantics:

$$\frac{}{\langle a, \lambda \rangle . P \overset{\delta_a^e}{\rightarrowtail} [P \mapsto \lambda]} \qquad \frac{\alpha \neq \delta_a^e}{\langle a, \lambda \rangle . P \overset{\alpha}{\rightarrowtail} []_{\mathbb{R}_{\geq 0}}}$$

$$\frac{P \overset{\alpha}{\rightarrowtail} \mathscr{P} \quad Q \overset{\alpha}{\rightarrowtail} \mathscr{Q} \quad (n\,\alpha) \notin L}{P \;||_L\; Q \overset{\alpha}{\rightarrowtail} (\mathscr{P} \otimes_{||_L} (\mathcal{X}_{\mathbb{R}}\, Q)) + ((\mathcal{X}_{\mathbb{R}}\, P) \otimes_{||_L} \mathscr{Q})}$$

$$\frac{P \overset{\alpha}{\rightarrowtail} \mathscr{P} \quad Q \overset{\alpha}{\rightarrowtail} \mathscr{Q} \quad (n\,\alpha) \in L}{P \;||_L\; Q \overset{\alpha}{\rightarrowtail} \mathscr{P} \otimes_{||_L} \mathscr{Q}}$$

Outline

PEPA - Hillston 1996

PEPA Operators

- *inaction*: **nil**
- *rated-action prefix*: $\langle a, \lambda \rangle.P$
- *choice*: $P + Q$
- *multi-party synchronization* $P \parallel_L Q$
- *constant*: X (where $X := P$)

PEPA - Hillston 1996

PEPA Operators

- *inaction*: **nil**
- *rated-action prefix*: $\langle a, \lambda \rangle.P$
- *choice*: $P + Q$
- *multi-party synchronization* $P \; ||_L \; Q$
- *constant*: X (where $X := P$)

The principle regulating the synchronization rate of PEPA processes is the so called *minimal rate*

- the rate of a synchronization is the MIN of the rates of synchronizing actions.

PEPA - Hillston 1996

From TIPP...

$$\frac{}{\langle a, \lambda \rangle . P \overset{\delta_a^e}{\rightarrowtail} [P \mapsto \lambda]} \qquad \frac{\alpha \neq \delta_a^e}{\langle a, \lambda \rangle . P \overset{\alpha}{\rightarrowtail} []_{\mathbb{R}_{\geq 0}}}$$

$$\frac{P \overset{\alpha}{\rightarrowtail} \mathscr{P} \quad Q \overset{\alpha}{\rightarrowtail} \mathscr{Q} \quad (n\,\alpha) \notin L}{P \mid\mid_L Q \overset{\alpha}{\rightarrowtail} (\mathscr{P} \otimes_{\mid\mid_L} (\mathcal{X}_{\mathbb{R}}\, Q)) + ((\mathcal{X}_{\mathbb{R}}\, P) \otimes_{\mid\mid_L} \mathscr{Q})}$$

$$\frac{P \overset{\alpha}{\rightarrowtail} \mathscr{P} \quad Q \overset{\alpha}{\rightarrowtail} \mathscr{Q} \quad (n\,\alpha) \in L}{P \mid\mid_L Q \overset{\alpha}{\rightarrowtail} \mathscr{P} \otimes_{\mid\mid_L} \mathscr{Q}}$$

... to PEPA

$$\frac{}{\langle a,\lambda\rangle.P \overset{\delta_a^e}{\rightarrowtail} [P \mapsto \lambda]} \qquad \frac{\alpha \neq \delta_a^e}{\langle a,\lambda\rangle.P \overset{\alpha}{\rightarrowtail} []_{\mathbb{R}_{\geq 0}}}$$

$$\frac{P \overset{\alpha}{\rightarrowtail} \mathscr{P} \quad Q \overset{\alpha}{\rightarrowtail} \mathscr{Q} \quad (n\,\alpha) \notin L}{P \;||_L\; Q \overset{\alpha}{\rightarrowtail} (\mathscr{P} \otimes_{||_L} (\mathcal{X}_{\mathbb{R}}\, Q)) + ((\mathcal{X}_{\mathbb{R}}\, P) \otimes_{||_L} \mathscr{Q})}$$

$$\frac{P \overset{\alpha}{\rightarrowtail} \mathscr{P} \quad Q \overset{\alpha}{\rightarrowtail} \mathscr{Q} \quad (n\,\alpha) \in L}{P \;||_L\; Q \overset{\alpha}{\rightarrowtail} \mathscr{P} \otimes_{||_L} \mathscr{Q} \cdot \frac{\mathrm{MIN}\{\oplus\mathscr{P}, \oplus\mathscr{Q}\}}{\oplus\mathscr{P} \cdot \oplus\mathscr{Q}}}$$

Outline

Stochastic CCS

CCS Operators

- *inaction*: **nil**,
- *rated-output-action prefix*: $\bar{a}^{\lambda}.P$,
- *passive-input-action prefix*: $a^{*_\omega}.P$,
- *choice*: $P + Q$, and
- *binary synchronization*: $P \mid Q$.

Stochastic CCS

CCS Operators

- *inaction*: **nil**,
- *rated-output-action prefix*: $\bar{a}^{\lambda}.P$,
- *passive-input-action prefix*: $a^{*\omega}.P$,
- *choice*: $P + Q$, and
- *binary synchronization*: $P \mid Q$.

- The duration of a synchronization is determined by the rate assigned to the participating output action.
- *Input actions* are annotated with *weights*, used for determining the probability that a specific input is selected.
- This approach is inspired by the notion of *passive actions* of EMPA and PEPA.

Stochastic CCS

Operational semantics:

$$\frac{}{\bar{a}^{\lambda}.P \overset{\delta^e_{\bar{a}}}{\rightarrowtail} [P \mapsto \lambda]} \qquad \frac{\alpha \neq \delta^e_{\bar{a}}}{\bar{a}^{\lambda}.P \overset{\alpha}{\rightarrowtail} []_{\mathbb{R}_{\geq 0}}}$$

$$\frac{}{a^{*\omega}.P \overset{\delta^e_{a}}{\rightarrowtail} [P \mapsto \omega]} \qquad \frac{\alpha \neq \delta^e_{a}}{a^{*\omega}.P \overset{\alpha}{\rightarrowtail} []_{\mathbb{N}_{\geq 0}}}$$

$$\frac{P \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \mathscr{P} \quad P \overset{\delta^e_{a}}{\rightarrowtail} \mathscr{P}_i \quad P \overset{\delta^e_{\bar{a}}}{\rightarrowtail} \mathscr{P}_o \quad Q \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \mathscr{Q} \quad Q \overset{\delta^e_{a}}{\rightarrowtail} \mathscr{Q}_i \quad Q \overset{\delta^e_{\bar{a}}}{\rightarrowtail} \mathscr{Q}_o}{P \mid Q \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \frac{(\mathscr{P} \otimes_{|} (\mathcal{X}_{\mathbb{R}_{\geq 0}} Q)) \cdot \oplus \mathscr{P}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{((\mathcal{X}_{\mathbb{R}_{\geq 0}} P) \otimes_{|} \mathscr{Q}) \cdot \oplus \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_i \otimes_{|} \mathscr{Q}_o}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_o \otimes_{|} \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i}}$$

Stochastic CCS

Synchronization rule:

$$\frac{P \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \mathscr{P} \quad P \overset{\delta^e_{a}}{\rightarrowtail} \mathscr{P}_i \quad P \overset{\delta^e_{\bar{a}}}{\rightarrowtail} \mathscr{P}_o \quad Q \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \mathscr{Q} \quad Q \overset{\delta^e_{a}}{\rightarrowtail} \mathscr{Q}_i \quad Q \overset{\delta^e_{\bar{a}}}{\rightarrowtail} \mathscr{Q}_o}{P \mid Q \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \frac{(\mathscr{P} \otimes_{|} (\mathcal{X}_{\mathbb{R}_{\geq 0}} Q)) \cdot \oplus \mathscr{P}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{((\mathcal{X}_{\mathbb{R}_{\geq 0}} P) \otimes_{|} \mathscr{Q}) \cdot \oplus \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_i \otimes_{|} \mathscr{Q}_o}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_o \otimes_{|} \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i}}$$

Stochastic CCS

Synchronization rule:

$$\frac{P \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \mathscr{P} \quad P \overset{\delta^e_{a}}{\rightarrowtail} \mathscr{P}_i \quad P \overset{\delta^e_{\bar{a}}}{\rightarrowtail} \mathscr{P}_o \quad Q \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \mathscr{Q} \quad Q \overset{\delta^e_{a}}{\rightarrowtail} \mathscr{Q}_i \quad Q \overset{\delta^e_{\bar{a}}}{\rightarrowtail} \mathscr{Q}_o}{P \mid Q \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \frac{(\mathscr{P} \otimes_{|} (\mathcal{X}_{\mathbb{R}_{\geq 0}} Q)) \cdot \oplus \mathscr{P}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{((\mathcal{X}_{\mathbb{R}_{\geq 0}} P) \otimes_{|} \mathscr{Q}) \cdot \oplus \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_i \otimes_{|} \mathscr{Q}_o}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_o \otimes_{|} \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i}}$$

1. the continuations of P after $\vec{a}$, in parallel with Q (rates are recomputed in order to take into account inputs in Q);

Stochastic CCS

Synchronization rule:

$$\frac{P \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \mathscr{P} \quad P \overset{\delta^e_{a}}{\rightarrowtail} \mathscr{P}_i \quad P \overset{\delta^e_{\bar{a}}}{\rightarrowtail} \mathscr{P}_o \quad Q \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \mathscr{Q} \quad Q \overset{\delta^e_{a}}{\rightarrowtail} \mathscr{Q}_i \quad Q \overset{\delta^e_{\bar{a}}}{\rightarrowtail} \mathscr{Q}_o}{P \mid Q \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \frac{(\mathscr{P} \otimes_{|} (\mathcal{X}_{\mathbb{R}_{\geq 0}} Q)) \cdot \oplus \mathscr{P}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{((\mathcal{X}_{\mathbb{R}_{\geq 0}} P) \otimes_{|} \mathscr{Q}) \cdot \oplus \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_i \otimes_{|} \mathscr{Q}_o}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_o \otimes_{|} \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i}}$$

1. the continuations of P after $\vec{a}$, in parallel with Q (rates are recomputed in order to take into account inputs in Q);
2. the continuations of Q after $\vec{a}$, in parallel with P (rates are recomputed in order to take into account inputs in P);

Stochastic CCS

Synchronization rule:

$$\frac{P \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \mathscr{P} \quad P \overset{\delta^e_{a}}{\rightarrowtail} \mathscr{P}_i \quad P \overset{\delta^e_{\bar{a}}}{\rightarrowtail} \mathscr{P}_o \quad Q \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \mathscr{Q} \quad Q \overset{\delta^e_{a}}{\rightarrowtail} \mathscr{Q}_i \quad Q \overset{\delta^e_{\bar{a}}}{\rightarrowtail} \mathscr{Q}_o}{P \mid Q \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \frac{(\mathscr{P} \otimes_{|} (\mathcal{X}_{\mathbb{R}_{\geq 0}} Q)) \cdot \oplus \mathscr{P}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{((\mathcal{X}_{\mathbb{R}_{\geq 0}} P) \otimes_{|} \mathscr{Q}) \cdot \oplus \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_i \otimes_{|} \mathscr{Q}_o}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_o \otimes_{|} \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i}}$$

1. the continuations of P after $\vec{a}$, in parallel with Q (rates are recomputed in order to take into account inputs in Q);
2. the continuations of Q after $\vec{a}$, in parallel with P (rates are recomputed in order to take into account inputs in P);
3. the continuations of P after a in parallel with the continuations of Q after $\bar{a}$, renormalized w.r.t. the total weight of inputs in Q;

Stochastic CCS

Synchronization rule:

$$\frac{P \stackrel{\delta^e_{\vec{a}}}{\rightarrowtail} \mathscr{P} \quad P \stackrel{\delta^e_{a}}{\rightarrowtail} \mathscr{P}_i \quad P \stackrel{\delta^e_{\bar{a}}}{\rightarrowtail} \mathscr{P}_o \quad Q \stackrel{\delta^e_{\vec{a}}}{\rightarrowtail} \mathscr{Q} \quad Q \stackrel{\delta^e_{a}}{\rightarrowtail} \mathscr{Q}_i \quad Q \stackrel{\delta^e_{\bar{a}}}{\rightarrowtail} \mathscr{Q}_o}{P \mid Q \stackrel{\delta^e_{\vec{a}}}{\rightarrowtail} \frac{(\mathscr{P} \otimes_{|} (\mathcal{X}_{\mathbb{R}_{\geq 0}} Q)) \cdot \oplus \mathscr{P}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{((\mathcal{X}_{\mathbb{R}_{\geq 0}} P) \otimes_{|} \mathscr{Q}) \cdot \oplus \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_i \otimes_{|} \mathscr{Q}_o}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_o \otimes_{|} \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i}}$$

1. the continuations of P after $\vec{a}$, in parallel with Q (rates are recomputed in order to take into account inputs in Q);
2. the continuations of Q after $\vec{a}$, in parallel with P (rates are recomputed in order to take into account inputs in P);
3. the continuations of P after a in parallel with the continuations of Q after $\bar{a}$, renormalized w.r.t. the total weight of inputs in Q;
4. the continuations of P after $\bar{a}$ in parallel with the continuations of Q after a, renormalized w.r.t. the total weight of inputs in P.

Stochastic CCS

Synchronization rule:

$$\frac{P \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \mathscr{P} \quad P \overset{\delta^e_{a}}{\rightarrowtail} \mathscr{P}_i \quad P \overset{\delta^e_{\bar{a}}}{\rightarrowtail} \mathscr{P}_o \quad Q \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \mathscr{Q} \quad Q \overset{\delta^e_{a}}{\rightarrowtail} \mathscr{Q}_i \quad Q \overset{\delta^e_{\bar{a}}}{\rightarrowtail} \mathscr{Q}_o}{P \mid Q \overset{\delta^e_{\vec{a}}}{\rightarrowtail} \frac{(\mathscr{P} \otimes_{|} (\mathcal{X}_{\mathbb{R}_{\geq 0}} Q)) \cdot \oplus \mathscr{P}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{((\mathcal{X}_{\mathbb{R}_{\geq 0}} P) \otimes_{|} \mathscr{Q}) \cdot \oplus \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_i \otimes_{|} \mathscr{Q}_o}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_o \otimes_{|} \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i}}$$

1. the continuations of P after $\vec{a}$, in parallel with Q (rates are recomputed in order to take into account inputs in Q);
2. the continuations of Q after $\vec{a}$, in parallel with P (rates are recomputed in order to take into account inputs in P);
3. the continuations of P after a in parallel with the continuations of Q after $\bar{a}$, renormalized w.r.t. the total weight of inputs in Q;
4. the continuations of P after $\bar{a}$ in parallel with the continuations of Q after a, renormalized w.r.t. the total weight of inputs in P.

Outline

A uniform account of quantitative PCs

To provide a uniform general account of the many *quantitative extensions* of PCs we have introduced a generalization of *RTS*s named FuTSs for *Function Labelled Transition Systems*.

- the transition relation associates to a state and a given transition label a function mapping each state into an element of a *commutative semi-ring* $\mathbb{C}$

A uniform account of quantitative PCs

To provide a uniform general account of the many *quantitative extensions* of PCs we have introduced a generalization of *RTS*s named FuTSs for *Function Labelled Transition Systems*.

- the transition relation associates to a state and a given transition label a function mapping each state into an element of a *commutative semi-ring* $\mathbb{C}$

A uniform account of quantitative PCs

To provide a uniform general account of the many *quantitative extensions* of PCs we have introduced a generalization of *RTS*s named FuTSs for *Function Labelled Transition Systems*.

- the transition relation associates to a state and a given transition label a function mapping each state into an element of a *commutative semi-ring* $\mathbb{C}$

By appropriately changing the set $\mathbb{C}$ we can capture different models of concurrent systems:

A uniform account of quantitative PCs

To provide a uniform general account of the many *quantitative extensions* of PCs we have introduced a generalization of *RTS*s named FuTSs for *Function Labelled Transition Systems*.

- the transition relation associates to a state and a given transition label a function mapping each state into an element of a *commutative semi-ring* $\mathbb{C}$

By appropriately changing the set $\mathbb{C}$ we can capture different models of concurrent systems:

- $\mathbb{B}$ consisting of the two boolean values *true* and *false* we can capture classical LTS;

A uniform account of quantitative PCs

To provide a uniform general account of the many *quantitative extensions* of PCs we have introduced a generalization of *RTS*s named FuTSs for *Function Labelled Transition Systems*.

- the transition relation associates to a state and a given transition label a function mapping each state into an element of a *commutative semi-ring* $\mathbb{C}$

By appropriately changing the set $\mathbb{C}$ we can capture different models of concurrent systems:

- $\mathbb{B}$ consisting of the two boolean values *true* and *false* we can capture classical LTS;
- $\mathbb{R}_{[0,1]}$ we do capture probabilistic models;

A uniform account of quantitative PCs

To provide a uniform general account of the many *quantitative extensions* of PCs we have introduced a generalization of *RTS*s named FuTSs for *Function Labelled Transition Systems*.

- the transition relation associates to a state and a given transition label a function mapping each state into an element of a *commutative semi-ring* $\mathbb{C}$

By appropriately changing the set $\mathbb{C}$ we can capture different models of concurrent systems:

- $\mathbb{B}$ consisting of the two boolean values *true* and *false* we can capture classical LTS;
- $\mathbb{R}_{[0,1]}$ we do capture probabilistic models;
- $\mathbb{R}_{\geq 0}$ we do capture stochastic models.

Semi-ring

A *semi-ring* is a set $\mathbb{S}$ equipped with two binary operations $+_{\mathbb{S}}$ (*sum*) and $\cdot_{\mathbb{S}}$ (*multiplication*) such that:

- $(\mathbb{S}, +_{\mathbb{S}})$ is a *commutative monoid* with neutral element $0_{\mathbb{S}} \in \mathbb{S}$;
- $(\mathbb{S}, \cdot_{\mathbb{S}})$ is a *monoid* with neutral element $1_{\mathbb{S}} \in \mathbb{S}$;
- multiplication distributes over sum
- $0_{\mathbb{S}}$ annihilates $\mathbb{S}$ with respect to moltiplication

Commutative Semi-ring

A *semi-ring* is a set $\mathbb{S}$ equipped with two binary operations $+_{\mathbb{S}}$ (*sum*) and $\cdot_{\mathbb{S}}$ (*multiplication*) such that:

- $(\mathbb{S}, +_{\mathbb{S}})$ is a *commutative monoid* with neutral element $0_{\mathbb{S}} \in \mathbb{S}$;
- $(\mathbb{S}, \cdot_{\mathbb{S}})$ is a commutative *monoid* with neutral element $1_{\mathbb{S}} \in \mathbb{S}$;
- multiplication distributes over sum
- $0_{\mathbb{S}}$ annihilates $\mathbb{S}$ with respect to moltiplication

Commutative Semi-ring

A *semi-ring* is a set $\mathbb{S}$ equipped with two binary operations $+_{\mathbb{S}}$ (*sum*) and $\cdot_{\mathbb{S}}$ (*multiplication*) such that:

- $(\mathbb{S}, +_{\mathbb{S}})$ is a *commutative monoid* with neutral element $0_{\mathbb{S}} \in \mathbb{S}$;
- $(\mathbb{S}, \cdot_{\mathbb{S}})$ is a commutative *monoid* with neutral element $1_{\mathbb{S}} \in \mathbb{S}$;
- multiplication distributes over sum
- $0_{\mathbb{S}}$ annihilates $\mathbb{S}$ with respect to moltiplication

Binary operation $/^{\mathbb{S}}$ is the *inverse* of $\cdot_{\mathbb{S}}$:

$$s_3 = s_1 /^{\mathbb{S}} s_2 \Leftrightarrow s_1 = s_2 \cdot_{\mathbb{S}} s_3 \quad (s_2 \neq 0_{\mathbb{S}})$$

FuTS: Function Labelled Transition Systems

Notations...

- $\mathbf{TF}(S, \mathbb{C})$ denote the set of *total* functions from S to $\mathbb{C}$
    - elements are ranged over by $\mathscr{P}, \mathscr{Q}, \mathscr{R}, \ldots$
    - $\mathbf{FTF}(S, \mathbb{C})$ denotes the set of total functions with *finite support*
- $[s_1 \mapsto \gamma_1, \ldots, s_m \mapsto \gamma_m]_{\mathbb{C}}$ denotes the function associating γ_i to s_i and $0_{\mathbb{C}}$
- $[]_{\mathbb{C}}$ denotes the $0_{\mathbb{C}}$ constant function
- functions in $\mathbf{TF}(S, \mathbb{C})$ can be composed with $+$:

$$(\mathscr{P} + \mathscr{Q})\, s \;=_{\mathrm{def}}\; (\mathscr{P}\, s) +_{\mathbb{C}} (\mathscr{Q}\, s)$$

- $\bigoplus \mathscr{P}$ denotes:

$$\bigoplus \mathscr{P}\, S' \;=_{\mathrm{def}}\; \sum_{s \in S'}{}_{\mathbb{C}} (\mathscr{P} s)$$

FuTS: Function Labelled Transition Systems

An *A-labelled function transition system* (FuTS) *over* $\mathbb{C}$ is a tuple $(S, A, \mathbb{C}, \rightarrowtail)$ where S is a countable, non-empty, set of *states*, A is a countable, non-empty, set of transition *labels*, $\mathbb{C}$ is a commutative semi-ring, and $\rightarrowtail \subseteq S \times A \times \mathbf{TF}(S, \mathbb{C})$ is the *transition relation*.

FuTS: Function Labelled Transition Systems

An *A-labelled function transition system* (FuTS) *over* $\mathbb{C}$ is a tuple $(S, A, \mathbb{C}, \rightarrowtail)$ where S is a countable, non-empty, set of *states*, A is a countable, non-empty, set of transition *labels*, $\mathbb{C}$ is a commutative semi-ring, and $\rightarrowtail \subseteq S \times A \times \mathbf{TF}(S, \mathbb{C})$ is the *transition relation*.

Transition relation $\rightarrowtail$ in a FuTS associates to each state s and transition label α a total function $(\mathscr{P}, \mathscr{Q}, \ldots)$ that assigns a value of a commutative semi-ring $\mathbb{C}$ to each process of the calculus.

FuTS: Function Labelled Transition Systems

An *A-labelled function transition system* (FuTS) *over* $\mathbb{C}$ is a tuple $(S, A, \mathbb{C}, \rightarrowtail)$ where S is a countable, non-empty, set of *states*, A is a countable, non-empty, set of transition *labels*, $\mathbb{C}$ is a commutative semi-ring, and $\rightarrowtail \subseteq S \times A \times \mathbf{TF}(S, \mathbb{C})$ is the *transition relation*.

Transition relation $\rightarrowtail$ in a FuTS associates to each state s and transition label α a total function $(\mathscr{P}, \mathscr{Q}, \ldots)$ that assigns a value of a commutative semi-ring $\mathbb{C}$ to each process of the calculus.

$s_1 \overset{\alpha}{\rightarrowtail} \mathscr{P}$ means that, for a generic state s_2:

- if $\mathscr{P}(s_2) = x$ $(\neq 0_{\mathbb{C}})$ then s_2 is reachable from s_2 via the execution of α
- if $\mathscr{P}(s_2) = 0$ then s_2 is not reachable from s_1 via α

FuTS: Function Labelled Transition Systems

Let $\mathcal{R} = (S, A, \mathbb{C}, \rightarrowtail)$ be a FuTS, then:

1. $\mathcal{R}$ is total if for all $s \in S$ and $\alpha \in A$ there exists $\mathscr{P} \in \mathbf{TF}(S, \mathbb{C})$ such that $s \overset{\alpha}{\rightarrowtail} \mathscr{P}$;
2. $\mathcal{R}$ is deterministic if for all $s \in S$, $\alpha \in A$, and $\mathscr{P}, \mathscr{Q} \in \mathbf{TF}(S, \mathbb{C})$ we have that the following holds: $s \overset{\alpha}{\rightarrowtail} \mathscr{P}, s \overset{\alpha}{\rightarrowtail} \mathscr{Q} \Longrightarrow \mathscr{P} = \mathscr{Q}$;
3. $\mathcal{R}$ is a finite support FuTS (FuTS_{FS} for short) if $\rightarrowtail \subseteq S \times A \times \mathbf{FTF}(S, \mathbb{C})$.

FuTS: Function Labelled Transition Systems

Let $\mathcal{R} = (S, A, \mathbb{C}, \rightarrowtail)$ be a FuTS, then:

1. $\mathcal{R}$ is total if for all $s \in S$ and $\alpha \in A$ there exists $\mathscr{P} \in \mathbf{TF}(S, \mathbb{C})$ such that $s \overset{\alpha}{\rightarrowtail} \mathscr{P}$;
2. $\mathcal{R}$ is deterministic if for all $s \in S$, $\alpha \in A$, and $\mathscr{P}, \mathscr{Q} \in \mathbf{TF}(S, \mathbb{C})$ we have that the following holds: $s \overset{\alpha}{\rightarrowtail} \mathscr{P}, s \overset{\alpha}{\rightarrowtail} \mathscr{Q} \Longrightarrow \mathscr{P} = \mathscr{Q}$;
3. $\mathcal{R}$ is a finite support FuTS (FuTS$_{FS}$ for short) if $\rightarrowtail \subseteq S \times A \times \mathbf{FTF}(S, \mathbb{C})$.

N.B. Deterministic FuTS can model *non-deterministic* behaviours!

A Labeled Transition System (LTS) is a triple $(S, A, \rightarrowtail)$ where:

- S is a countable set of states.
- A is a countable set of transition-labeling actions.
- $\rightarrowtail \; \subseteq S \times A \times S$ is a transition relation.

A Labeled Transition System (LTS) is a triple $(S, A, \rightarrowtail)$ where:

- S is a countable set of states.
- A is a countable set of transition-labeling actions.
- $\rightarrowtail \; \subseteq S \times A \times S$ is a transition relation.

A LTS is a total and deterministic FuTS over $\mathbb{B}$ where:

- $\mathbb{B} = \{\bot, \top\}$ is the Boolean algebra
- $s \overset{a}{\rightarrowtail} \mathscr{P}$: $\mathscr{P}(s') = \top \Leftrightarrow s \overset{a}{\rightarrowtail} s'$

An action-labeled discrete-time Markov chain (ADTMC) is a triple $(S, A, \rightarrowtail)$ where:

- S is a countable set of states.
- A is a countable set of transition-labeling actions.
- $\rightarrowtail \subseteq S \times A \times \mathbb{R}_{(0,1]} \times S$ is a transition relation.
  - $(s, a, p_1, s'), (s, a, p_2, s') \in \rightarrowtail \implies p_1 = p_2$.
  - $\sum \{\!| p \in \mathbb{R}_{(0,1]} \mid \exists a \in A, s' \in S. (s, a, p, s') \in \rightarrowtail |\!\} \in \{0, 1\}$.

An action-labeled discrete-time Markov chain (ADTMC) is a triple $(S, A, \rightarrowtail)$ where:

- S is a countable set of states.
- A is a countable set of transition-labeling actions.
- $\rightarrowtail \ \subseteq S \times A \times \mathbb{R}_{(0,1]} \times S$ is a transition relation.
  - $(s, a, p_1, s'), (s, a, p_2, s') \in \rightarrowtail \implies p_1 = p_2$.
  - $\sum \{\!| \, p \in \mathbb{R}_{(0,1]} \mid \exists a \in A, s' \in S. \, (s, a, p, s') \in \rightarrowtail \, |\!\} \in \{0, 1\}$.

An ADTMC is a total and deterministic FuTS $\mathbb{R}_{[0,1]}$ such that

- $\sum_{s \overset{a}{\rightarrowtail} \mathscr{P}} \sum_{s' \in S} \mathscr{P}(s') \in \{0, 1\}$
- $s \overset{a}{\rightarrowtail} \mathscr{P}, \ \mathscr{P}s' = p > 0 \Leftrightarrow (s, a, p, s') \in \rightarrowtail$

An action-labeled continuous-time Markov chain (ACTMC) is a triple $(S, A, \rightarrowtail)$ where:

- S is a countable set of states.
- A is a countable set of transition-labeling actions.
- $\rightarrowtail\ \subseteq S \times A \times \mathbb{R}_{>0} \times S$ is a transition relation.
  - $(s, a, \lambda_1, s'), (s, a, \lambda_2, s') \in \rightarrowtail\ \Longrightarrow \lambda_1 = \lambda_2$.

An action-labeled continuous-time Markov chain (ACTMC) is a triple $(S, A, \rightarrowtail)$ where:

- S is a countable set of states.
- A is a countable set of transition-labeling actions.
- $\rightarrowtail \subseteq S \times A \times \mathbb{R}_{>0} \times S$ is a transition relation.
  - $(s, a, \lambda_1, s'), (s, a, \lambda_2, s') \in \rightarrowtail \implies \lambda_1 = \lambda_2$.

An ACTMC is a total and deterministic FuTS over $\mathbb{R}_{\geq 0}$ such that:

- $s \xrightarrow{a} \mathscr{P}$: $\mathscr{P} = v > 0 \Leftrightarrow (s, a, v, s') \in \rightarrowtail$

Outline

Behavioral Equivalences on FuTS

Behavioral Equivalences on FUTS

Let $\mathcal{R} = (S, A, \mathbb{C}, \rightarrowtail)$ be a FUTS over $\mathbb{C}$. A trace w for $\mathcal{R}$ is a finite sequence of transition labels in A^*, where $w = \varepsilon$ denotes the empty sequence while operation "$\_ \circ \_$" denotes sequence concatenation.

Behavioral Equivalences on FuTS

Let $\mathcal{R} = (S, A, \mathbb{C}, \rightarrowtail)$ be a FuTS over $\mathbb{C}$. A trace w for $\mathcal{R}$ is a finite sequence of transition labels in A^*, where $w = \varepsilon$ denotes the empty sequence while operation "$\_ \circ \_$" denotes sequence concatenation.

Let $\mathcal{R} = (S, A, \mathbb{C}, \rightarrowtail)$ be a FuTS $\mathbb{C}$ and M be a lattice. A measure function for $\mathcal{R}$ is a function $\mathcal{M}_M : S \times A^* \times 2^S \rightarrow M$.

Behavioral Equivalences on FuTS

Trace Equivalence

Let $\mathcal{R} = (S, A, \mathbb{C}, \rightarrowtail)$ be a FuTS over $\mathbb{C}$ and $\mathcal{M}_M$ be a measure function for $\mathcal{R}$.
Two states $s_1, s_2 \in S$ are $\mathcal{M}_M$-trace equivalent iff for all traces $w \in A^*$:

$$\mathcal{M}_M(s_1, w, S) \; = \; \mathcal{M}_M(s_2, w, S)$$

Behavioral Equivalences on FuTS

Trace Equivalence

Let $\mathcal{R} = (S, A, \mathbb{C}, \rightarrowtail)$ be a FuTS over $\mathbb{C}$ and $\mathcal{M}_M$ be a measure function for $\mathcal{R}$.
Two states $s_1, s_2 \in S$ are $\mathcal{M}_M$-trace equivalent iff for all traces $w \in A^*$:

$$\mathcal{M}_M(s_1, w, S) \;=\; \mathcal{M}_M(s_2, w, S)$$

Bisimulation Equivalence

Let $\mathcal{R} = (S, A, \mathbb{C}, \rightarrowtail)$ be a FuTS over $\mathbb{C}$ and $\mathcal{M}_M$ be a measure function for $\mathcal{R}$. An equivalence relation $\mathcal{B}$ over S is an $\mathcal{M}_M$-bisimulation iff, whenever $(s_1, s_2) \in \mathcal{B}$, for all traces $w \in A^*$ and equivalence classes $C \in S/\mathcal{B}$:

$$\mathcal{M}_M(s_1, w, C) \;=\; \mathcal{M}_M(s_2, w, C)$$

Two states $s_1, s_2 \in S$ are $\mathcal{M}_M$-bisimilar iff there exists an $\mathcal{M}_M$-bisimulation $\mathcal{B}$ over S such that $(s_1, s_2) \in \mathcal{B}$.

Measure Function for LTSs

Let $\mathcal{R} = (S, A, \mathbb{B}, \rightarrowtail)$ be a total and deterministic FuTS over $\mathbb{B}$.
Function $\mathcal{M}_{\mathbb{B}} : S \times A^* \times 2^S \to \mathbb{B}$ for $\mathcal{R}$ is inductively defined as follows:

$$\mathcal{M}_{\mathbb{B}}(s, w, S') = \begin{cases} \bigvee_{s' \in S} \mathscr{P}_{s,a}(s') \wedge \mathcal{M}_{\mathbb{B}}(s', w', S') & \text{if } w = \alpha \circ w', s \overset{\alpha}{\rightarrowtail} \mathscr{P} \\ \top & \text{if } w = \varepsilon \text{ and } s \in S' \\ \bot & \text{if } w = \varepsilon \text{ and } s \notin S' \end{cases}$$

Measure Function for LTSs

Let $\mathcal{R} = (S, A, \mathbb{B}, \rightarrowtail)$ be a total and deterministic FuTS over $\mathbb{B}$.
Function $\mathcal{M}_{\mathbb{B}} : S \times A^* \times 2^S \to \mathbb{B}$ for $\mathcal{R}$ is inductively defined as follows:

$$\mathcal{M}_{\mathbb{B}}(s, w, S') = \begin{cases} \bigvee_{s' \in S} \mathscr{P}_{s,a}(s') \wedge \mathcal{M}_{\mathbb{B}}(s', w', S') & \text{if } w = \alpha \circ w', s \overset{\alpha}{\rightarrowtail} \mathscr{P} \\ \top & \text{if } w = \varepsilon \text{ and } s \in S' \\ \bot & \text{if } w = \varepsilon \text{ and } s \notin S' \end{cases}$$

$\mathcal{M}_{\mathbb{B}}(s, w, S') = \top$ if and only if s reaches an element in S' with trace w.

Measure Function for LTSs

Let $\mathcal{R} = (S, A, \mathbb{B}, \rightarrowtail)$ be a total and deterministic FuTS over $\mathbb{B}$.
Function $\mathcal{M}_{\mathbb{B}} : S \times A^* \times 2^S \to \mathbb{B}$ for $\mathcal{R}$ is inductively defined as follows:

$$\mathcal{M}_{\mathbb{B}}(s, w, S') = \begin{cases} \bigvee_{s' \in S} \mathscr{P}_{s,a}(s') \wedge \mathcal{M}_{\mathbb{B}}(s', w', S') & \text{if } w = \alpha \circ w', s \overset{\alpha}{\rightarrowtail} \mathscr{P} \\ \top & \text{if } w = \varepsilon \text{ and } s \in S' \\ \bot & \text{if } w = \varepsilon \text{ and } s \notin S' \end{cases}$$

$\mathcal{M}_{\mathbb{B}}(s, w, S') = \top$ if and only if s reaches an element in S' with trace w.

Trace and Bisimulation equivalences on LTS coincide with $\mathcal{M}_{\mathbb{B}}$-trace and $\mathcal{M}_{\mathbb{B}}$-bisimulation on FuTS over $\mathbb{B}$.

Measure Function for ADTMC

Let $\mathcal{R} = (S, A, \mathbb{R}_{[0,1]}, \rightarrowtail)$ be a total and deterministic FuTS over $\mathbb{R}_{[0,1]}$.
Function $\mathcal{M}_{\mathbb{R}_{[0,1]}} : S \times A^* \times 2^S \to \mathbb{R}_{[0,1]}$ for $\mathcal{R}$ is inductively defined by:

$$\mathcal{M}_{\mathbb{R}_{[0,1]}}(s, w, S') = \begin{cases} \sum_{s' \in S} \mathscr{P}(s') \cdot \mathcal{M}_{\mathbb{R}_{[0,1]}}(s', w', S') & \text{if } w = \alpha \circ w', s \overset{\alpha}{\rightarrowtail} \mathscr{P} \\ 1 & \text{if } w = \varepsilon \text{ and } s \in S' \\ 0 & \text{if } w = \varepsilon \text{ and } s \notin S' \end{cases}$$

$\mathcal{M}_{\mathbb{R}_{[0,1]}}(s, w, S')$ gives the probability that s reaches an element in S' with trace w.

Measure Function for ADTMC

Let $\mathcal{R} = (S, A, \mathbb{R}_{[0,1]}, \rightarrowtail)$ be a total and deterministic FuTS over $\mathbb{R}_{[0,1]}$. Function $\mathcal{M}_{\mathbb{R}_{[0,1]}} : S \times A^* \times 2^S \rightarrow \mathbb{R}_{[0,1]}$ for $\mathcal{R}$ is inductively defined by:

$$\mathcal{M}_{\mathbb{R}_{[0,1]}}(s, w, S') = \begin{cases} \sum_{s' \in S} \mathscr{P}(s') \cdot \mathcal{M}_{\mathbb{R}_{[0,1]}}(s', w', S') & \text{if } w = \alpha \circ w', s \overset{\alpha}{\rightarrowtail} \mathscr{P} \\ 1 & \text{if } w = \varepsilon \text{ and } s \in S' \\ 0 & \text{if } w = \varepsilon \text{ and } s \notin S' \end{cases}$$

$\mathcal{M}_{\mathbb{R}_{[0,1]}}(s, w, S')$ gives the probability that s reaches an element in S' with trace w.

Trace and Bisimulation equivalences on ADTMC coincides with $\mathcal{M}_{\mathbb{R}_{[0,1]}}$-trace and $\mathcal{M}_{\mathbb{R}_{[0,1]}}$-bisimulation on FuTS over $\mathbb{R}_{[0,1]}$.

Measure Function for ACTMC

Let $\mathcal{R} = (S, A, \mathbb{R}_{\geq 0}, \rightarrowtail)$ be a total and deterministic FuTS over $\mathbb{R}_{\geq 0}$. The end-to-end measure function $\mathcal{M}_{\mathrm{ete}} : S \times A^* \times 2^S \to [\mathbb{R}_{\geq 0} \to \mathbb{R}_{[0,1]}]$ for $\mathcal{R}$ is inductively defined as follows:

$$\mathcal{M}_{\mathrm{ete}}(s, \alpha, S')(t) = \begin{cases} \int_0^t \mathrm{E}(s) \cdot \mathrm{e}^{-\mathrm{E}(s)\cdot x} \cdot \sum_{s' \in S} \frac{\mathscr{P}(s')}{\mathrm{E}(s)} \cdot \mathcal{M}_{\mathrm{ete}}(s', \alpha', S')(t - x)\, \mathrm{d}x & \text{if } \alpha = a \circ \alpha' \text{ and } \mathrm{E}(s) > 0 \\ 1 & \text{if } \alpha = \varepsilon \text{ and } s \in S' \\ 0 & \text{if } \alpha = \varepsilon \text{ and } s \notin S' \text{ or } \\ & \alpha \neq \varepsilon \text{ and } \mathrm{E}(s) = 0 \end{cases}$$

Measure Function for ACTMC

Let $\mathcal{R} = (S, A, \mathbb{R}_{\geq 0}, \rightarrowtail)$ be a total and deterministic FuTS over $\mathbb{R}_{\geq 0}$. The end-to-end measure function $\mathcal{M}_{\mathrm{ete}} : S \times A^* \times 2^S \to [\mathbb{R}_{\geq 0} \to \mathbb{R}_{[0,1]}]$ for $\mathcal{R}$ is inductively defined as follows:

$$\mathcal{M}_{\mathrm{ete}}(s, \alpha, S')(t) = \begin{cases} \int_0^t \mathrm{E}(s) \cdot \mathrm{e}^{-\mathrm{E}(s) \cdot x} \cdot \sum_{s' \in S} \frac{\mathscr{P}(s')}{\mathrm{E}(s)} \cdot \mathcal{M}_{\mathrm{ete}}(s', \alpha', S')(t-x)\, \mathrm{d}x & \text{if } \alpha = a \circ \alpha' \text{ and } \mathrm{E}(s) > 0 \\ 1 & \text{if } \alpha = \varepsilon \text{ and } s \in S' \\ 0 & \text{if } \alpha = \varepsilon \text{ and } s \notin S' \text{ or } \\ & \alpha \neq \varepsilon \text{ and } \mathrm{E}(s) = 0 \end{cases}$$

$\mathcal{M}_{\mathrm{ete}}(s, w, S')(t)$ gives the probability that s reaches an element in S' with trace w within t time units.

Measure Function for ACTMC

Let $\mathcal{R} = (S, A, \mathbb{R}_{\geq 0}, \rightarrowtail)$ be a total and deterministic FuTS over $\mathbb{R}_{\geq 0}$. The end-to-end measure function $\mathcal{M}_{\mathrm{ete}} : S \times A^* \times 2^S \rightarrow [\mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{[0,1]}]$ for $\mathcal{R}$ is inductively defined as follows:

$$\mathcal{M}_{\mathrm{ete}}(s, \alpha, S')(t) = \begin{cases} \int_0^t \mathrm{E}(s) \cdot \mathrm{e}^{-\mathrm{E}(s) \cdot x} \cdot \sum_{s' \in S} \frac{\mathscr{P}(s')}{\mathrm{E}(s)} \cdot \mathcal{M}_{\mathrm{ete}}(s', \alpha', S')(t - x)\, \mathrm{d}x & \text{if } \alpha = a \circ \alpha' \text{ and } \mathrm{E}(s) > 0 \\ 1 & \text{if } \alpha = \varepsilon \text{ and } s \in S' \\ 0 & \text{if } \alpha = \varepsilon \text{ and } s \notin S' \text{ or } \\ & \alpha \neq \varepsilon \text{ and } \mathrm{E}(s) = 0 \end{cases}$$

$\mathcal{M}_{\mathrm{ete}}(s, w, S')(t)$ gives the probability that s reaches an element in S' with trace w within t time units.

Trace and Bisimulation equivalences on ACTMC coincide with $\mathcal{M}_{\mathrm{ete}}$-trace and $\mathcal{M}_{\mathrm{ete}}$-bisimulation on FuTS over $\mathbb{R}_{\geq 0}$.

Additional work done

FuTSs have been used to give stochastic semantics of a large class of Stochastic Process Calculi also with non determinism:

- EMPA
- Stochastic-π
- StoKlaim
- Language for Interactive Markov Chains (IM)

Additional work done

FuTSs have been used to give stochastic semantics of a large class of Stochastic Process Calculi also with non determinism:

- EMPA
- Stochastic-π
- StoKlaim
- Language for Interactive Markov Chains (IM)

A general notion of *testing equivalence* have been defined on FuTS:

- FuTSs are composed with a test;
- function $\mathcal{M}$ gives a measure of the test-success.

Outline

Summing Up

We have:

Summing Up

We have:

- introduced first *RTS*s then FuTS and have used them as the basic model for defining stochastic behaviours of a number of process calculi:
  - A language for CTMC
  - TIPP
  - PEPA
  - Stochastic CCS

Summing Up

We have:

- introduced first *RTS*s then FuTS and have used them as the basic model for defining stochastic behaviours of a number of process calculi:
  - A language for CTMC
  - TIPP
  - PEPA
  - Stochastic CCS
- identified a uniform formalization of standard equivalences:
  - trace and bisimulation equivalences
  - based on the notion of *measure functions*
  - capture usual notions on classical computational models

Future Work

- General FuTS over generic Commutativa Semirings;
- Taxonomy of existing behavioural relations;
- Consider FuTS with explicit representation of non-determinism;
- FuTS and co-algebras;
- FuTS and weighted automata.

Thank you for your attention!

References

BERNARDO, M., DE NICOLA, R., AND LORETI, M.
Uniform labeled transition systems for nondeterministic, probabilistic, and stochastic processes.
LNCS, vol. 6084, 2010. Springer, 35–56.

DE NICOLA, R., LATELLA, D., LORETI, M., AND MASSINK, M.
Rate-based Transition Systems for Stochastic Process Calculi.
LNCS, vol. 5556, 2009. Springer-Verlag, 435–446.

DE NICOLA, R., LATELLA, D., LORETI, M., AND MASSINK, M.
On a Uniform Framework for the Definition of Stochastic Process Languages.
LNCS, vol. 5825, 2009. Springer, 9–25.

DE NICOLA, R., LATELLA, D., LORETI, M., AND MASSINK, M.
A Uniform Definition of Stochastic Process Calculi.
Survey paper, submitted for publication. If interested send me an email.