Compositional modelling of concurrent systems and their quantitative evaluation

Rocco De Nicola

Università degli Studi di Firenze
visiting IMT - Lucca

Camerino
September 15, 2010

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

Outline...

Formal Methods for Reactive Systems

To deal with reactive systems and guarantee their correct behavior in all possible environment, we need:

1. To study mathematical models for the formal description and analysis of concurrent programs.

Formal Methods for Reactive Systems

To deal with reactive systems and guarantee their correct behavior in all possible environment, 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.

Formal Methods for Reactive Systems

To deal with reactive systems and guarantee their correct behavior in all possible environment, 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

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

Operational Semantics

To each process built using the above operators we associate an LTS by relying on structural induction to define the meaning of each operator.

Definition (Inference Systems)

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

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

In our case for a generic operator *op* we shall have one or more rules like:

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

Presentations of Labelled Transition Systems

Process Algebra as denotations of LTS

- LTS are represented by terms of process algebras.
- Terms are interpreted via operational semantics as LTS.

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.

Stochastic Process Algebras for quantitative properties

A number of stochastic process algebras have been proposed in the last two decades. These are based on:

1. Labeled Transition Systems (LTS)
   - for providing compositional semantics of languages
   - for describing *qualitative properties*
2. Continuous Time Markov Chains (CTMC)
   - for analysing *quantitative properties*

Stochastic Process Algebras for quantitative properties

A number of stochastic process algebras have been proposed in the last two decades. These are based on:

1. Labeled Transition Systems (LTS)
    - for providing compositional semantics of languages
    - for describing *qualitative properties*
2. Continuous Time Markov Chains (CTMC)
    - for analysing *quantitative properties*

Semantics of stochastic calculi have been provided by resorting to variants of the Structured Operational Semantics (SOS) approach but:

- there is no general framework for modelling the different formalisms
- it is rather difficult to appreciate differences and similarities of such semantics.

Stochastic Process Algebras - incomplete list

- TIPP (N. Glotz, U. Herzog, M. Rettelbach - 1993)
- Stochastic π-calculus (C. Priami - 1995, later with P. Quaglia)
- PEPA (J. Hillston - 1996)
- EMPA (M. Bernardo, R. Gorrieri - 1998)
- IMC (H. Hermanns - 2002)
- ...
- StoKlaim
- MarCaSPiS
- ...

More Calculi will come: Besides qualitative aspects of distributed systems it more and more important that performance and dependability be addressed to deal, e.g., with issues related to quality of service.

Common ingredients of Stochastic PA

Randomized Actions

- It is assumed that action execution takes time
- Execution times is described by means of random variables
- Random Variables are assumed to be exponentially distributed
- Random Variables are fully characterised by their rates.

Common ingredients of Stochastic PA

Randomized Actions

- It is assumed that action execution takes time
- Execution times is described by means of random variables
- Random Variables are assumed to be exponentially distributed
- Random Variables are fully characterised by their rates.

Properties of Exponential Distributions

If X is *exponentially distributed* with *parameter* $\lambda \in \mathbf{R}_{>0}$:

- $\mathbb{P}\{X \leq d\} = 1 - e^{-\lambda \cdot d}$, for $d \geq 0$
- The average duration of X is $\frac{1}{\lambda}$; the variance of X is $\frac{1}{\lambda^2}$
- *Memory-less*: $\mathbb{P}\{X \leq t + d \mid X > t\} \; = \; \mathbb{P}\{X \leq d\}$

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
- $\mathbf{R} : \mathcal{S} \times \mathcal{S} \rightarrow \mathrm{R}_{\geq 0}$, the rate matrix

Stochastic process calculi

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

Stochastic process calculi

- A CTMC is associated to each process term;
- CTMC 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;
- CTMC 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;
- CTMC 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:

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

$$\textbf{rec } X \, . \, \alpha.X \mid \textbf{rec } X \, . \, \alpha.X = \textbf{rec } X \, . \, \alpha.X$$

Stochastic process calculi

- A CTMC is associated to each process term;
- CTMC 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:

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

$$\textbf{rec}\ X \,.\, \alpha^{\lambda}.X \mid \textbf{rec}\ X \,.\, \alpha^{\lambda}.X \quad \neq \quad \textbf{rec}\ X \,.\, \alpha^{\lambda}.X$$

Stochastic process calculi

- A CTMC is associated to each process term;
- CTMC 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:

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

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

Semantics of stochastic process calculi

We introduce a variant of Rate Transition Systems (RTS), proposed by Klin and Sassone(FOSSACS 2008), and use them for defining stochastic behaviour of a few process algebras.

Semantics of stochastic process calculi

We introduce a variant of Rate Transition Systems (RTS), proposed by Klin and Sassone(FOSSACS 2008), and use them for defining stochastic behaviour of a few process algebras.

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

We introduce a variant of Rate Transition Systems (RTS), proposed by Klin and Sassone(FOSSACS 2008), and use them for defining stochastic behaviour of a few process algebras.

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.

- Our variant of RTS associates terms and actions to functions from terms to rates
- The *apparent rate* approach, originally developed by Hillston for multi-party synchronisation (à la CSP), is generalized to deal "appropriately" also with binary synchronisation (à la CCS).

Semantics of stochastic process calculi

Stochastic semantics of process calculi is defined by means of a transition relation $\longrightarrow$ 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 $\longrightarrow$ 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

Semantics of stochastic process calculi

Stochastic semantics of process calculi is defined by means of a transition relation $\longrightarrow$ 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 $(S, A, \longrightarrow)$ where:

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

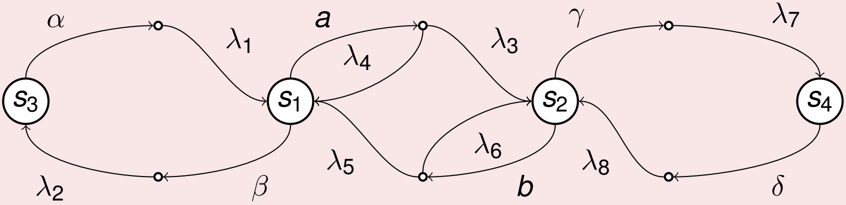
Rate transition systems

Definition

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

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

An example of RTS



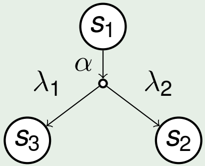
Rate transition systems

Definition

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

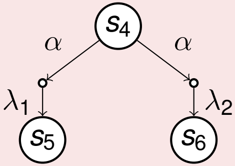
- $\mathcal{R}$ is *functional* if and only if for each $s \in S$, $\alpha \in A$, $\mathscr{P}$ and $\mathscr{Q}$ we have: $s \xrightarrow{\alpha} \mathscr{P}, s \xrightarrow{\alpha} \mathscr{Q} \Longrightarrow \mathscr{P} = \mathscr{Q}$
- $\mathcal{R}$ is *image finite* if and only if for each $s \in S$, $\alpha \in A$ and $\mathscr{P}$ such that $s \xrightarrow{\alpha} \mathscr{P}$ we have: $\{s' | \mathscr{P}(s') > 0\}$ is finite

A functional RTS



that leads to a CTMC.

A general RTS

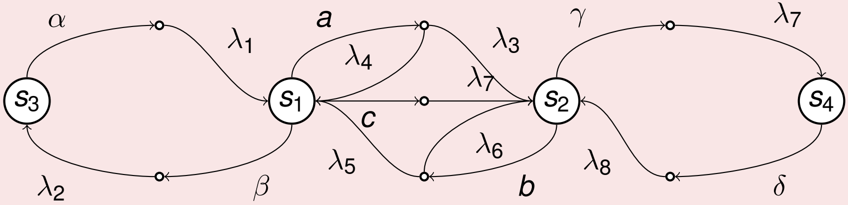


that leads to a CTM Dec. Proc.

A translation from an RTS to a CTMC

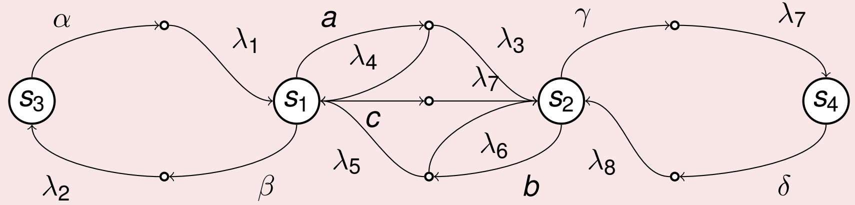
A translation from an RTS to a CTMC

An RTS:

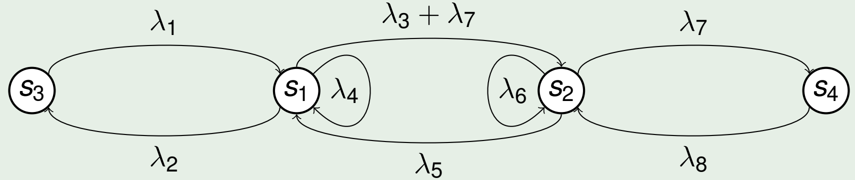


A translation from an RTS to a CTMC

An RTS:

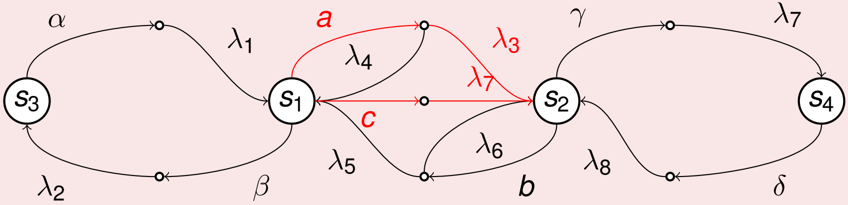


The corresponding CTMC:

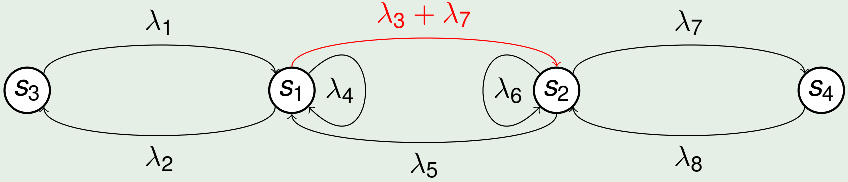


A translation from an RTS to a CTMC

An RTS:



The corresponding CTMC:



Strong Markovian Bisimilarity

Definition (Bisimulation)

Given a generic CTMC $(S, \mathbf{R})$

- An equivalence relation $\mathcal{E}$ on S is a Markovian bisimulation on S if and only if for all $(s_1, s_2) \in \mathcal{E}$ and for all equivalence classes $C \in S_{/\mathcal{E}}$ the following condition holds: $\mathbf{R}[s_1, C] \; = \; \mathbf{R}[s_2, C]$.

Definition (Bisimilarity)

Given a generic CTMC $(S, \mathbf{R})$

- Two states $s_1, s_2 \in S$ are strongly Markovian bisimilar, written $s_1 \sim_M s_2$, if and only if there exists a Markovian bisimulation $\mathcal{E}$ on S with $(s_1, s_2) \in \mathcal{E}$.

Rate aware bisimulation

Definition (Rate Aware Bisimilarity)

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

- An equivalence relation $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{S}$ is a *rate aware* bisimulation if and only if, for all $(s_1, s_2) \in \mathcal{E}$, and $\underline{S} \in \mathcal{S}_{/\mathcal{E}}$, and for all α and $\mathscr{P}$:
$$s_1 \xrightarrow{\alpha} \mathscr{P} \Longrightarrow \exists \mathscr{Q} : \; s_2 \xrightarrow{\alpha} \mathscr{Q} \wedge \mathscr{P}(\underline{S}) = \mathscr{Q}(\underline{S})$$
- Two states $s_1, s_2 \in S$ are *rate aware bisimilar* ($s_1 \sim s_2$) if there exists a rate aware bisimulation $\mathcal{E}$ such that $(s_1, s_2) \in \mathcal{E}$.

Theorem

Let $\mathcal{R} = (\mathcal{S}, A, \longrightarrow)$*, for each* $A' \subseteq A$ *and for each* $s_1, s_2 \in \mathcal{S}$ *and* $(S, \mathbf{R}) = CTMC[\{s_1, s_2\}, A']$*:* $s_1 \sim s_2 \Longrightarrow s_1 \sim_M s_2$

Rate aware bisimilarity and *strong bisimilarity* coincide if actions identity is not considered.

PEPA: Performance Process Algebra

PEPA Systems

PEPA systems are the result of *components* interaction via *activities*:

- Components reflect the behaviour of relevant parts of the system,
- activities model the actions components do perform.

PEPA Activities

Each PEPA activity consists of a pair (α, λ) where:

- α symbolically denotes the performed action;
- $\lambda > 0$ is the rate of the (negative) *exponential* distribution.

PEPA Syntax

If $\mathcal{A}$ is a set of *actions*, ranged over by $\alpha, \alpha', \alpha_1, \ldots$, then $\mathcal{P}_{PEPA}$ is the set of process terms $P, P', P_1, \ldots$ defined by:

$$P ::= (\alpha, \lambda).P \mid P + P \mid P \parallel_L P \mid P/L \mid A$$

PEPA Stochastic semantics...

$$\frac{}{(\alpha,\lambda).P \xrightarrow{\alpha} [P \mapsto \lambda]} \ (\textsc{Act}) \qquad \frac{\alpha \neq \beta}{(\alpha,\lambda).P \xrightarrow{\beta} \emptyset} \ (\emptyset\text{-}\textsc{Act})$$

$$\frac{P \xrightarrow{\alpha} \mathscr{P} \quad Q \xrightarrow{\alpha} \mathscr{Q}}{P + Q \xrightarrow{\alpha} \mathscr{P} + \mathscr{Q}} \ (\textsc{Sum}) \qquad \frac{P \xrightarrow{\alpha} \mathscr{P} \quad Q \xrightarrow{\alpha} \mathscr{Q} \quad \alpha \notin L}{P \mid\mid_L Q \xrightarrow{\alpha} \mathscr{P} \mid\mid_L \chi_Q + \chi_P \mid\mid_L \mathscr{Q}} \ (\textsc{Int})$$

$$\frac{P \xrightarrow{\alpha} \mathscr{P} \quad Q \xrightarrow{\alpha} \mathscr{Q} \quad \alpha \in L}{P \mid\mid_L Q \xrightarrow{\alpha} \mathscr{P} \mid\mid_L \mathscr{Q} \cdot \frac{\min\{\oplus\mathscr{P}, \oplus\mathscr{Q}\}}{\oplus\mathscr{P} \cdot \oplus\mathscr{Q}}} \ (\textsc{Coop})$$

$$\frac{P \xrightarrow{\alpha} \mathscr{P} \quad \alpha \notin L}{P/L \xrightarrow{\alpha} \mathscr{P}/L} \ (\text{P-}\textsc{Hide}) \qquad \frac{\alpha \in L}{P/L \xrightarrow{\alpha} \emptyset} \ (\emptyset\text{-}\textsc{Hide})$$

$$\frac{P \xrightarrow{\tau} \mathscr{P}_\tau \quad \forall \alpha \in L.P \xrightarrow{\alpha} \mathscr{P}_\alpha}{P/L \xrightarrow{\tau} \mathscr{P}_\tau/L + \sum_{\alpha \in L} \mathscr{P}_\alpha/L} \ (\textsc{Hide}) \qquad \frac{P \xrightarrow{\alpha} \mathscr{P} \quad A \triangleq P}{A \xrightarrow{\alpha} \mathscr{P}} \ (\textsc{Call})$$

PEPA Stochastic semantics...

Prefixes and Sums

$$\frac{}{(\alpha,\lambda).P \xrightarrow{\alpha} [P \mapsto \lambda]} \ (\textsc{Act}) \qquad \frac{\alpha \neq \beta}{(\alpha,\lambda).P \xrightarrow{\beta} \emptyset} \ (\emptyset\text{-}\textsc{Act})$$

$$\frac{P \xrightarrow{\alpha} \mathscr{P} \quad Q \xrightarrow{\alpha} \mathscr{Q}}{P + Q \xrightarrow{\alpha} \mathscr{P} + \mathscr{Q}} \ (\textsc{Sum})$$

PEPA Stochastic semantics...

Prefixes and Sums

$$\frac{}{(\alpha,\lambda).P \xrightarrow{\alpha} [P \mapsto \lambda]} \ (\textsc{Act}) \qquad \frac{\alpha \neq \beta}{(\alpha,\lambda).P \xrightarrow{\beta} \emptyset} \ (\emptyset\text{-}\textsc{Act})$$

$$\frac{P \xrightarrow{\alpha} \mathscr{P} \quad Q \xrightarrow{\alpha} \mathscr{Q}}{P + Q \xrightarrow{\alpha} \mathscr{P} + \mathscr{Q}} \ (\textsc{Sum})$$

Interleaving and Multiparty Synchronization

$$\frac{P \xrightarrow{\alpha} \mathscr{P} \quad Q \xrightarrow{\alpha} \mathscr{Q} \quad \alpha \notin L}{P \,||_L\, Q \xrightarrow{\alpha} \mathscr{P} \,||_L\, \chi_Q + \chi_P \,||_L\, \mathscr{Q}} \qquad \frac{P \xrightarrow{\alpha} \mathscr{P} \quad Q \xrightarrow{\alpha} \mathscr{Q} \quad \alpha \in L}{P \,||_L\, Q \xrightarrow{\alpha} \mathscr{P} \,||_L\, \mathscr{Q} \cdot \frac{\min\{\oplus\mathscr{P}, \oplus\mathscr{Q}\}}{\oplus\mathscr{P} \cdot \oplus\mathscr{Q}}}$$

Other Stochastic Process Algebras considered

- EMPA
- TIPP
- Stochastic CCS
- Stochastic π-calculus
- IMC
- StoKlaim
- MarCaSPiS

Tools

For all these formalisms we have built prototypes to perform systems analysis

More quantitative process algebras

In the last twenty years we have seen many variants of process algebras that have been introduces to capture different non-qualitative aspects of concurrent systems:

- probability
- Time
- Causality
- Security
- ...

More quantitative process algebras

In the last twenty years we have seen many variants of process algebras that have been introduces to capture different non-qualitative aspects of concurrent systems:

- probability
- Time
- Causality
- Security
- . . .

A new challenge

Can we find a general model of processes that encompasses all the many process algebraic models introduced to describe and asses qualitative and quantitative properties of concurrent reactive systems?

ULTRAS: Uniform Labelled Transition Systems

Uniform Labelled Transition Systems

Let D be a complete partial order (cpo) with least element $\perp$, a Uniform Transition System on D, that we call $D-$ ULTRAS, is a triple

$$(S, A, \longrightarrow)$$

where:

- S is a set of states,
- A a set of transition labels,
- $\longrightarrow$ a subset of $S \times A \times [S \rightarrow D]$.

ULTRAS: Uniform Labelled Transition Systems

Traces:

Let $\mathcal{U} = (S, A, \longrightarrow)$ be a D-ULTRAS. A trace α for $\mathcal{U}$ is a finite sequence of transition labels in A^*, where $\alpha = \varepsilon$ denotes the empty sequence while operation "$\_ \circ \_$" denotes sequence concatenation.

Measuring function:

Let $\mathcal{U} = (S, A, \longrightarrow)$ be a D-ULTRAS and M be a lattice. A measuring function for $\mathcal{U}$ is a function $\mathcal{M}_M : S \times A^* \times 2^S \to M$.

Trace equivalence

Let $\mathcal{U} = (S, A, \longrightarrow)$ be a D-ULTRAS and $\mathcal{M}_M$ be a measuring function for $\mathcal{U}$. We say that $s_1, s_2 \in S$ are $\mathcal{M}_M$-trace equivalent, written $s_1 \sim_{\mathrm{Tr},\mathcal{M}_M} s_2$, iff for all traces $\alpha \in A^*$:

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

Behavioural equivalences for ULTRAS

Bisimulation

Let $\mathcal{U} = (S, A, \longrightarrow)$ be a D-ULTRAS and $\mathcal{M}_M$ be a measuring function for $\mathcal{U}$. An equivalence relation $\mathcal{B}$ over S is an $\mathcal{M}_M$-bisimulation iff, whenever $(s_1, s_2) \in \mathcal{B}$, then for all traces $\alpha \in A^*$ and equivalence classes $C \in S/\mathcal{B}$:

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

Bisimilarity

We say that $s_1, s_2 \in S$ are $\mathcal{M}_M$-bisimilar, written $s_1 \sim_{\mathrm{B},\mathcal{M}_M} s_2$, iff there exists an $\mathcal{M}_M$-bisimulation $\mathcal{B}$ over S such that $(s_1, s_2) \in \mathcal{B}$.

Behavioural equivalences for ULTRAS

Testing equivalence:

Let $(S, A, \longrightarrow)$ be a D-ULTRAS and $\mathcal{M}_M$ be a measuring function for $\mathcal{U}$.

Two states s_1, s_2 are *testing equivalent* if and only if, for each trace $\alpha \in A^*$ and $A' \subseteq A$:

$$\mathcal{M}_M(s_1, \alpha, M_{set}(A')) = \mathcal{M}_M(s_2, \alpha, M_{set}(A'))$$

Must Sets:

Let $\mathcal{R} = (S, A, \longrightarrow)$ be a D-ULTRAS, $A' \subseteq A$, $s \in S$ and $a \in A$:

- s must a iff $\exists\, \mathscr{P} \neq \lambda x.\bot$ such that $s \xrightarrow{a} \mathscr{P}$
- s Must A' iff $\exists\, a \in A'$ such that s must a
- $M_{set}(A') = \{s \in S \mid \text{s } \mathit{Must} \text{ A'}\}$

LTS as ULTRAS

Labelled Transition System:

A labeled transition system (LTS for short) is a triple $(S, A, \longrightarrow)$ where:

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

A Labelled Transition System can be rendered as a $\mathbb{B}$-ULTRAS where:

- $\mathbb{B}$ is the set of boolean values $(\{\top, \bot\})$;

Measuring function:

$$\mathcal{M}_{\mathbb{B}}(s, \alpha, S') = \begin{cases} 1 & s \text{ reaches } S' \text{ with } \alpha \\ 0 & \text{otherwise} \end{cases}$$

ADTMC as ULTRAS

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

- S is a countable set of states.
- A is a countable set of transition-labeling actions.
- $\longrightarrow \subseteq S \times A \times \mathbb{R}_{(0,1]} \times S$ is a (probabilistic) transition relation.

ADTMC is a functional $\mathbb{R}_{[0,1]}$-ULTRAS in which:

$$\sum_{s \xrightarrow{a} \mathcal{D}} \sum_{s' \in S} \mathcal{D}(s') \in \{0, 1\}$$

Measuring function:

$\mathcal{M}_{[0,1]}(s, \alpha, S') =$ probability to reach a state in S' from s with trace α.

ADTMC as ULTRAS

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

- S is a countable set of states.
- A is a countable set of transition-labeling actions.
- $\longrightarrow \subseteq S \times A \times \mathbf{R}_{>0} \times S$ is a transition relation.

An ACTMC is a functional $\mathbf{R}_{\geq 0}$-ULTRAS.

Measuring function:

$\mathcal{M}_{\mathbf{R}_{\geq 0} \to [0,1]}(s, \alpha, S') =$ probability distribution to reach a state in S' from s with trace α.

Correspondence results

Trace, *Testing* and *Bisimulation* equivalences classically defined on LTS, ADTMC and ACTMC coincide with the ones induced by:

- $\mathbb{B}$ – ULTRAS and $\mathcal{M}_{\mathbb{B}}$;
- $[0,1]$ – ULTRAS and $\mathcal{M}_{[0,1]}$;
- $\mathbf{R}_{\geq 0}$ – ULTRAS and $\mathcal{M}_{\mathbf{R}_{\geq 0} \to [0,1]}$.

Summing Up

We have:

- introduced Rate Transition Systems and have used them as the basic model for defining stochastic behaviours of processes.
- introduced a natural notion of bisimulation over RTS that agrees with Markovian bisimulation.
- shown how RTS can be used to provide the stochastic operational semantics of PEPA (... and other SPA).
- introduced ULTRASas more general models of quantitative systems
- defined equivalence relations over ULTRAS
- shown that ULTRAScan be used for modelling other semantics (non-deterministic, stochastic, probabilitistic,. . .)

Thank you for your attention!