Cosmos Overview

Benoît Barbot

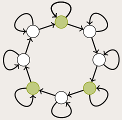
LACL, Université Paris-Est Créteil

ETR Aout 2017

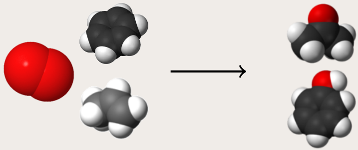
Probabilistic Systems

Inherent Stochastic Systems

Telecommunication Protocols using Randomness

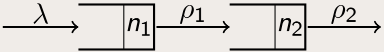


Chemical Reaction

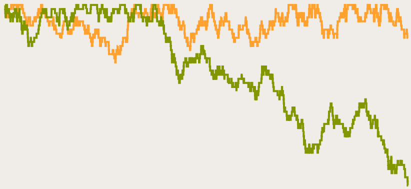


Modelling of Unknown Parts of Systems

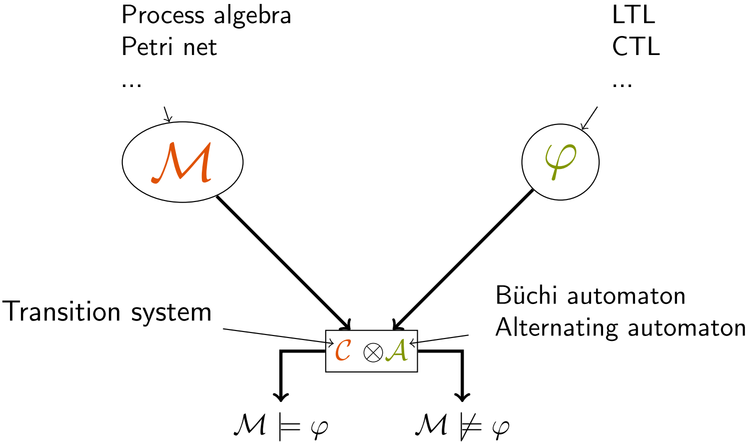
Waiting Queues



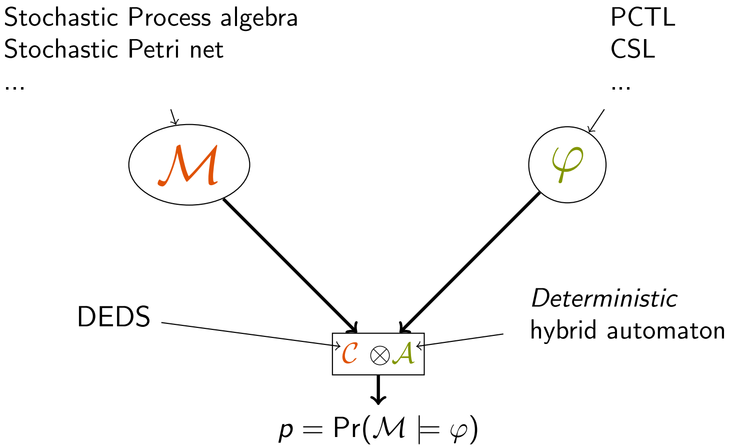
Banking System



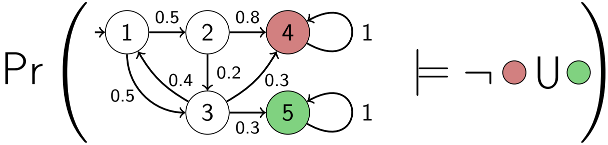
Model Checking



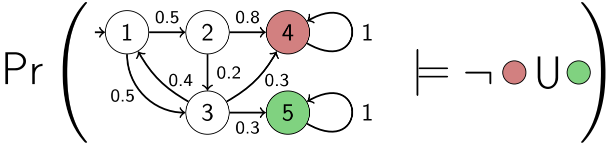
Model Checking for Stochastic Systems



Numerical vs Statistical Approaches



Numerical vs Statistical Approaches



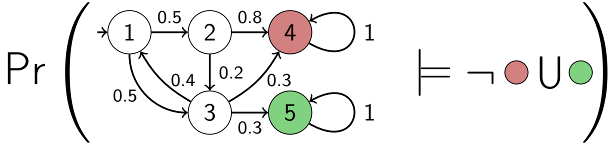
Numerical Model Checking

$$\begin{pmatrix} -1 & 0.5 & 0.5 & 0 & 0 \\ 0 & -1 & 0.2 & 0.8 & 0 \\ 0.4 & 0 & -1 & 0.3 & 0.3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\Downarrow$$

$$x_1 = 0.2368$$

Numerical vs Statistical Approaches



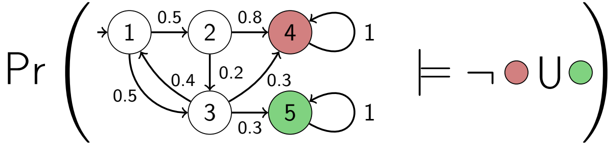
Statistical Model Checking

$1 \to 2 \to 4$ x
$1 \to 3 \to 1 \to 3 \to 5$ ✓
$1 \to 3 \to 4$ x
$1 \to 3 \to 1 \to 3 \to 1 \to 2 \to 4$ x
$1 \to 2 \to 3 \to 5$ ✓
...

⇓

$$x_1 \in [0.19, 0.26]$$

Numerical vs Statistical Approaches



Numerical Model Checking

$$\begin{pmatrix} -1 & 0.5 & 0.5 & 0 & 0 \\ 0 & -1 & 0.2 & 0.8 & 0 \\ 0.4 & 0 & -1 & 0.3 & 0.3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\Downarrow$$

$$x_1 = 0.2368$$

Statistical Model Checking

$1 \to 2 \to 4$ ✗
$1 \to 3 \to 1 \to 3 \to 5$ ✓
$1 \to 3 \to 4$ ✗
$1 \to 3 \to 1 \to 3 \to 1 \to 2 \to 4$ ✗
$1 \to 2 \to 3 \to 5$ ✓
...

$$\Downarrow$$

$$x_1 \in [0.19, 0.26]$$

Numerical vs Statistical Approaches

Numerical approach

- Precise value (but prone to numerical errors)
- Strong probabilistic hypotheses
- Memory space proportional to the size of the stochastic process

Numerical vs Statistical Approaches

Numerical approach

- Precise value (but prone to numerical errors)
- Strong probabilistic hypotheses
- Memory space proportional to the size of the stochastic process

Statistical approach

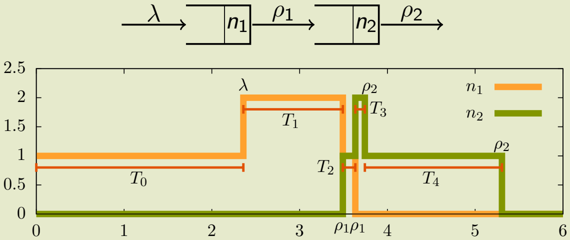
- Confidence interval: probabilistic framing
- Small memory space
- Easy to parallelise
- Weak probabilistic hypotheses (only an operational semantic)
- Requires fully stochastic models
- Rare events problem

Discrete Event Dynamic System(DEDS)

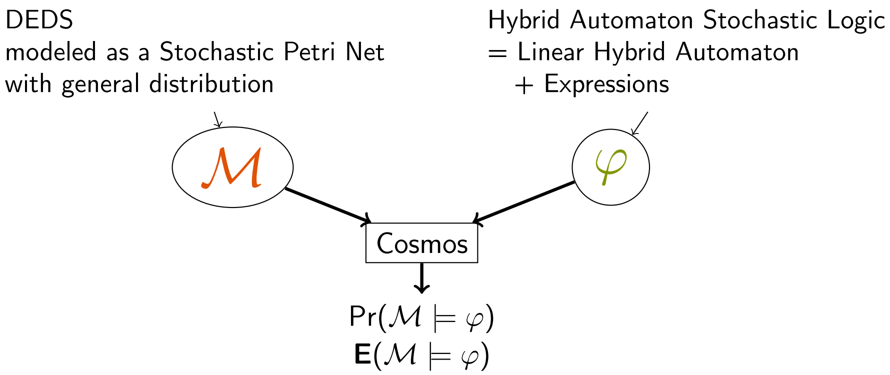
DEDS: $(S, S_0, E, \delta, (E_n)_0^\infty, (T_n)_0^\infty)$

- A discrete set of state S, initial state is a random variable (RV) $S_0 \in S$
- A set of events E
- A transition function $\delta : S \times E \to S$
- A sequence of RV $(E_n)_0^\infty$. The sequence of states is $S_{n+1} = \delta(S_n, E_n)$.
- A sequence of RV in $\mathbb{R}^+$: $(T_n)_0^\infty$

DEDS realisation example



Cosmos

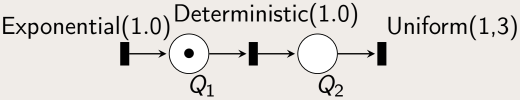


Synchronisation

- DEDS generates random timed words.
- Automaton tries to read the word.
- Expressions are evaluated on the variable of the automaton.

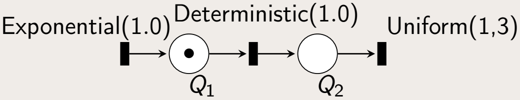
Generalised Stochastic Petri Net

Example Description (Tandem Queues)



Generalised Stochastic Petri Net

Example Description (Tandem Queues)

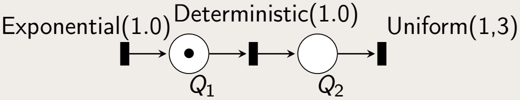


Description

- A Petri net; defines state space, events and transitions.
- After a transition is enabled the time before firing is distributed according to the distribution.
- The next event is the transition with smallest firing time.

Generalised Stochastic Petri Net

Example Description (Tandem Queues)



Description

- A Petri net; defines state space, events and transitions.
- After a transition is enabled the time before firing is distributed according to the distribution.
- The next event is the transition with smallest firing time.

Extensions

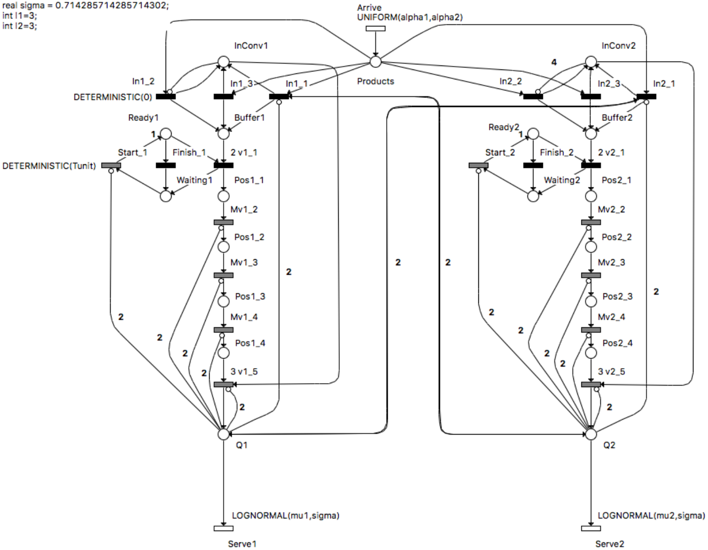
- Petri net with inhibitor arcs, marking dependant valuation.
- Coloured Petri net

Petri Net Demo

System Example

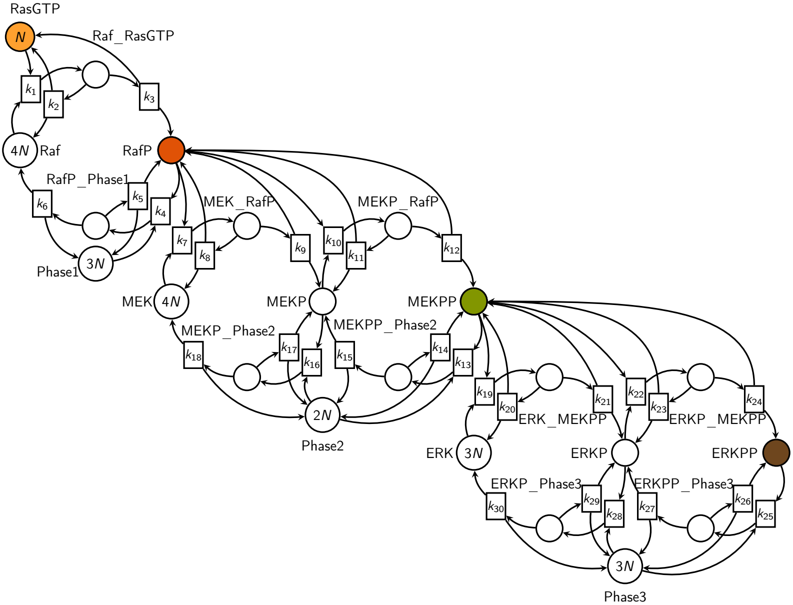
Flexible Manufacturing Systems

```
real alpha1=0.45;
real alpha2=0.55;
real Tunit=0.5;
real mu1 = 0.714285714285714302;
real mu2 = 0.357142857142857151;
real sigma = 0.714285714285714302;
int l1=3;
int l2=3;
```



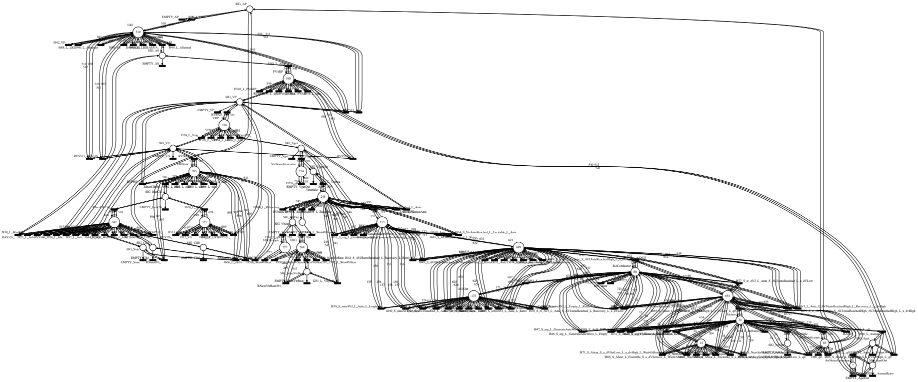
System Example

Molecular Signalling Pathway



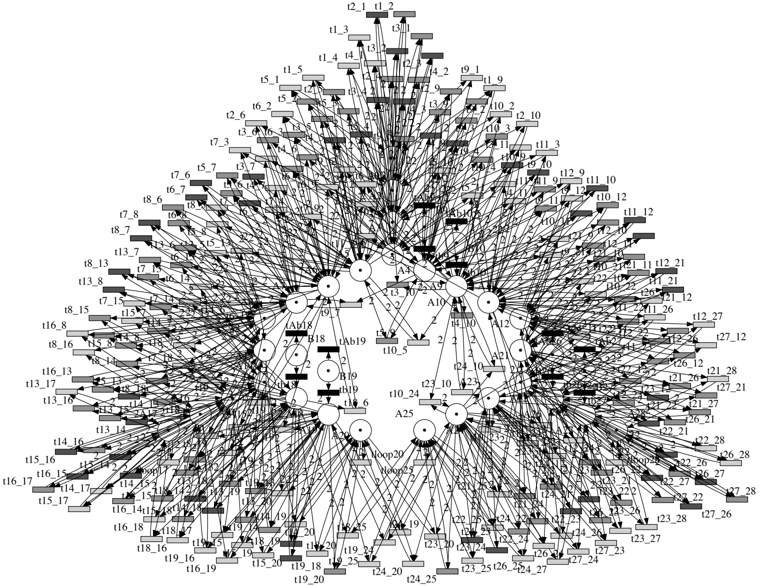
System Example

Human Heart and Pacemaker system



System Example

Computation with DNA



Specification Language

Expressivity

Given a set of trajectories obtained by simulation, what can we compute ?

$$s_1 \xrightarrow{T_1,E_1} s_2 \xrightarrow{T_2,E_2} s_3 \xrightarrow{T_3,E_3} s_4 \xrightarrow{T_4,E_4} \cdots \xrightarrow{T_{n-1},E_{n-1}} s_n$$

Specification Language

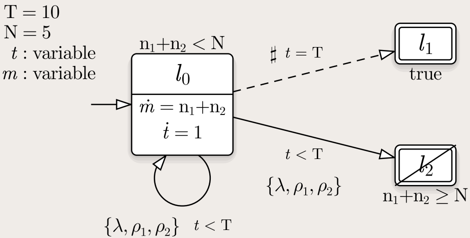
Expressivity

Given a set of trajectories obtained by simulation, what can we compute ?

$$s_1 \xrightarrow{T_1, E_1} s_2 \xrightarrow{T_2, E_2} s_3 \xrightarrow{T_3, E_3} s_4 \xrightarrow{T_4, E_4} \cdots \xrightarrow{T_{n-1}, E_{n-1}} s_n$$

Linear Hybrid Automaton (LHA)

- An automaton labelled by set of DEDS events or #.
- A set of variables with flows.
- Assignment of variable.
- Linear guard and invariant.



LHA Semantic

Two kinds of transitions

- Synchronised transition
  $\Rightarrow$ DEDS and LHA change state at the same time
- Autonomous transition (#)
  $\Rightarrow$ only the LHA changes location, as soon as the guard is satisfied

LHA Semantic

Two kinds of transitions

- Synchronised transition
  $\Rightarrow$ DEDS and LHA change state at the same time
- Autonomous transition (#)
  $\Rightarrow$ only the LHA changes location, as soon as the guard is satisfied

Time behaviours

- Flows of clocks are expressions on the state of the DEDS
  $\Rightarrow$ Piece-wise linear
- Guards are linear expressions on variables
  $\Rightarrow$ guard satisfaction boils down to solving linear system

LHA Semantic

Two kinds of transitions

- Synchronised transition
  $\Rightarrow$ DEDS and LHA change state at the same time
- Autonomous transition (#)
  $\Rightarrow$ only the LHA changes location, as soon as the guard is satisfied

Time behaviours

- Flows of clocks are expressions on the state of the DEDS
  $\Rightarrow$ Piece-wise linear
- Guards are linear expressions on variables
  $\Rightarrow$ guard satisfaction boils down to solving linear system

Determined

- One initial location
- Final locations
- The automaton is deterministic

Hybrid Automata Stochastic Logic (HASL)

HASL formula

- An LHA
- An expression over variables of the automaton, to compute complex indexes on the *accepted* path.

Hybrid Automata Stochastic Logic (HASL)

HASL formula

- An LHA
- An expression over variables of the automaton, to compute complex indexes on the *accepted* path.

Formula construction

$$\underbrace{AVG\,(}_{\text{Probabilistic operator}}\ \overbrace{\text{Integral(}}^{\text{Path operator}}\ \underbrace{t - m}_{\text{linear expression}}\ )\ \overbrace{+\,\text{Last(}}^{\text{Path operator}}\ m\,))$$

Hybrid Automata Stochastic Logic (HASL)

HASL formula

- An LHA
- An expression over variables of the automaton, to compute complex indexes on the *accepted* path.

Formula construction

$$\underbrace{AVG\ (}_{\text{Probabilistic operator}}\ \overbrace{\text{Integral(}}^{\text{Path operator}}\ \underbrace{t - m}_{\text{linear expression}}\ )\ \overbrace{+\ \text{Last(}}^{\text{Path operator}}\ m\ ))$$

Probabilistic operator

- *PROB*
- $AVG(X)$
- $PDF(X, step, min, max)$
- $CDF(X, step, min, max)$

Path operator

- Last(x)
- Integral(x)
- Mean(x)
- Min(x) / Max(x)

Hasl evaluation

Synchronisation

- Simulation of the GSPN
- Synchronisation of the LHA
- Trajectory is accepted if a final state is reached
- Trajectory is rejected if LHA fail to synchronise

Hasl evaluation

Synchronisation

- Simulation of the GSPN
- Synchronisation of the LHA
- Trajectory is accepted if a final state is reached
- Trajectory is rejected if LHA fail to synchronise

Hasl expression

- Linear expression evaluated after each step of simulation
- Path expression evaluated along the path
- Probabilistic operator evaluated on set of trajectories

Hasl evaluation

Synchronisation

- Simulation of the GSPN
- Synchronisation of the LHA
- Trajectory is accepted if a final state is reached
- Trajectory is rejected if LHA fail to synchronise

Hasl expression

- Linear expression evaluated after each step of simulation
- Path expression evaluated along the path
- Probabilistic operator evaluated on set of trajectories

Trajectories are not stored !

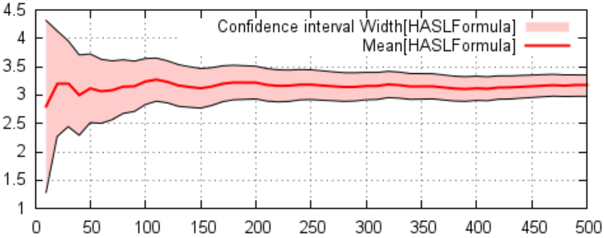
No dynamic allocation of memory !

Confidence Interval

Confidence Interval

Given a random variable X and a confidence level $1 - \varepsilon$, an estimator of the expected value of X returns a confidence interval I if

$$\Pr(\mathbf{E}(X) \in I) \geq 1 - \varepsilon$$



Three parameters: confidence level, confidence interval width and number of samples. Two of them have to be fixed.

Cosmos 1/2

Description: a command-line tool

- Input model: a Generalised Stochastic Petri Net
- Input specification: HASL formulas
- Input: Statistical Parameters
- Output: Probabilistic framing of values of HASL formulas

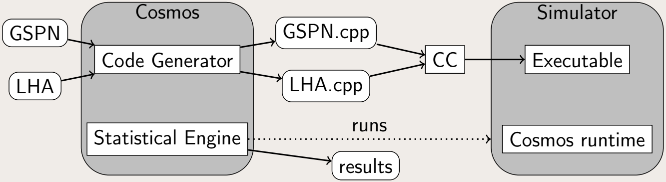
Cosmos 1/2

Description: a command-line tool

- Input model: a Generalised Stochastic Petri Net
- Input specification: HASL formulas
- Input: Statistical Parameters
- Output: Probabilistic framing of values of HASL formulas

Architecture

- Contains 25 Kloc of C/C++ and OCaml under GPLv2
- Generates code implementing the synchronisation GSPN/LHA
- Distributes simulation



Cosmos 2/2

Features

- Static and Sequential statistical methods: Chernoff-Hoeffding, Chow-Robbins, Gaussian, SPRT
- Several input formats: GrML, Marcie, PNML, Prism
- Several compatible editing tools: Coloane, GreatSPN Editor, Snoopy
- Plain and coloured Petri nets
- Fast thanks to structural analysis of Petri net and code generation
- Low memory footprint
- Various possible outputs

Cosmos 2/2

Features

- Static and Sequential statistical methods: Chernoff-Hoeffding, Chow-Robbins, Gaussian, SPRT
- Several input formats: GrML, Marcie, PNML, Prism
- Several compatible editing tools: Coloane, GreatSPN Editor, Snoopy
- Plain and coloured Petri nets
- Fast thanks to structural analysis of Petri net and code generation
- Low memory footprint
- Various possible outputs

Extensions

- Handling of Rare Events with importance sampling
- Uniform sampling for time automata
- Hardware in the loop simulation
- Simulation of hybrid models: Simulink

Conclusion

- Fast and lightweight statistical model checker.
- Rich classes of input models.
- Rich specification language
- Modular and open source $\rightarrow$ easy to hack

Conclusion

- Fast and lightweight statistical model checker.
- Rich classes of input models.
- Rich specification language
- Modular and open source $\rightarrow$ easy to hack

Download

`http://www.lsv.ens-cachan.fr/Software/cosmos/`