CSL$^{\mathsf{LHA}}$: an Expressive Language for Statistical Verification of Stochastic Models

Paolo Ballarini[1] Hilal Djafri[2] Marie Duflot[1]
Serge Haddad[2] Nihal Pekergin[2]

[1]LACL, Univesrité Paris-Est Créteil
[2]LSV, ENS-Cachan

what is this presentation about?

- we introduce a new methodology for the automatic verification of stochastic models
- principal features are:
  - highly expressive formalism which allows for capturing sophisticated dynamics/measures of a model
  - making stochastic model checking leaning towards performance evaluation
  - formalism uses Linear Hybrid Automata as a means to expresses the properties/measures of interest
  - the verification/estimation procedure is statistical (simulation based), thus not affected by state-space-explosion problem

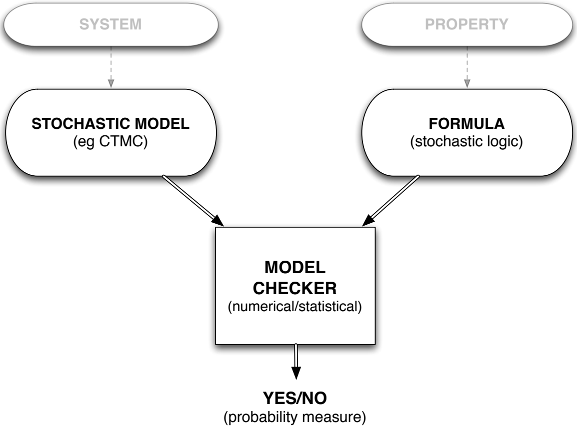
Outline

Stochastic Logics

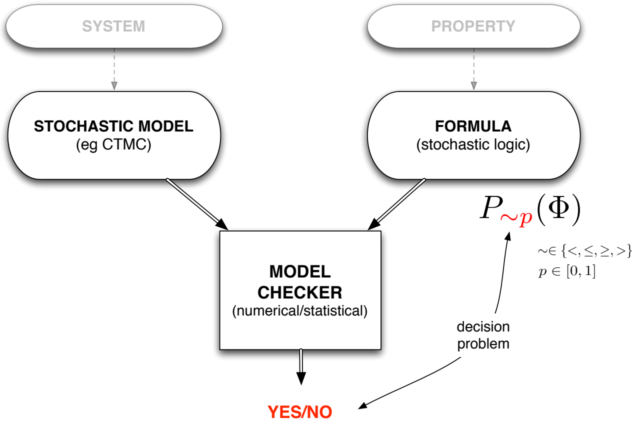
CSL^{LHA}

Software support for CSL^{LHA}

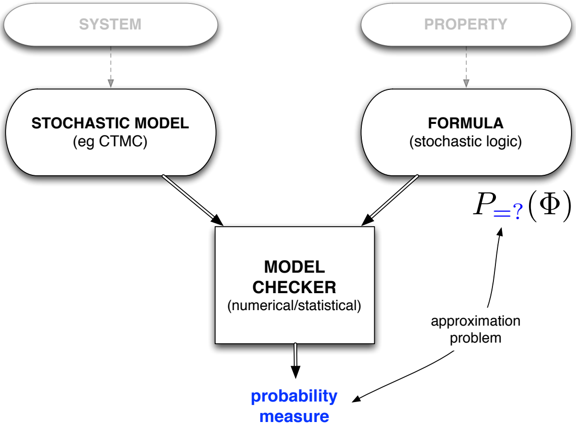
Stochastic Model Checking



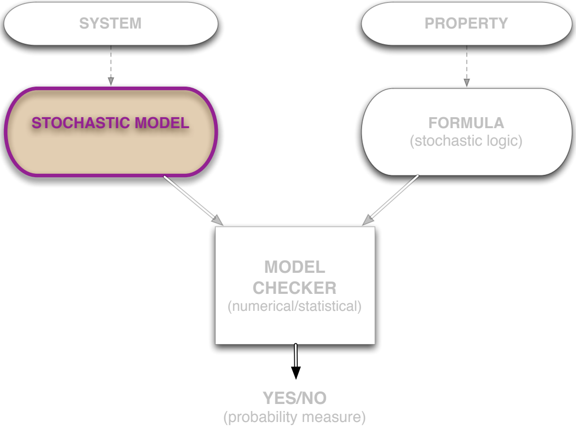
Stochastic Model Checking



Stochastic Model Checking



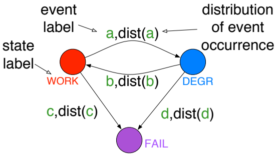
Stochastic Model Checking



Discrete Event Stochastic Process

an abstraction whereby a real system is represented in terms of:

- **states:** enumerable set of states $S = \{s_0, \ldots, s_n \ldots\}$,
- **events**: finite set of events $E = \{e_0, \ldots, e_m\}$; occurrence time is driven by a probability distribution

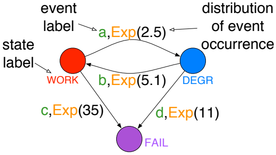


Discrete Event
Stochastic Process

Discrete Event Stochastic Process

an abstraction whereby a real system is represented in terms of:

- **states:** enumerable set of states $S = \{s_0, \ldots, s_n \ldots\}$,
- **events**: finite set of events $E = \{e_0, \ldots, e_m\}$; occurrence time is driven by a probability distribution

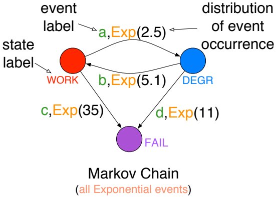


Markov Chain
(all Exponential events)

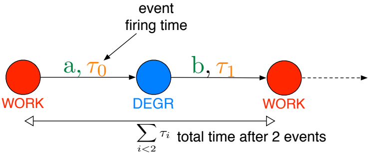
Discrete Event Stochastic Process

an abstraction whereby a real system is represented in terms of:

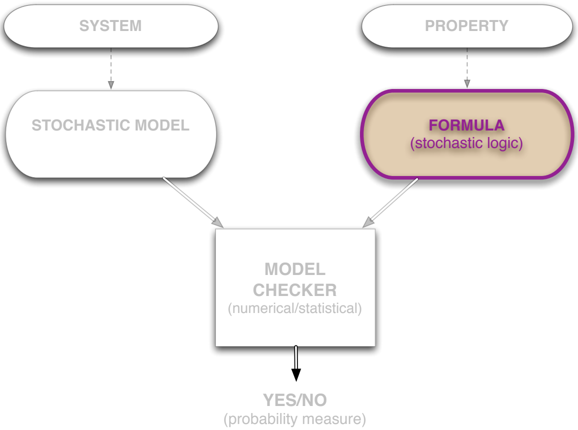
- **states:** enumerable set of states $S = \{s_0, \ldots, s_n \ldots\}$,
- **events**: finite set of events $E = \{e_0, \ldots, e_m\}$; occurrence time is driven by a probability distribution



- **path:** a (possibly infinite) sequence of events occurrences



Stochastic Model Checking



Stochastic Logic

basic idea: to extend reasoning of classical temporal logic (LTL/CTL) to stochastic models models

- **syntax based**: properties are expressed in terms of a formula
  - Continuous Stochastic Logic (**CSL**); [Aziz 2000]
  - action-state Continuous Stochastic Logic (**asCSL**); [Baier *et al.* 2004]
  - Continuous Stochastic Reward Logic (**CSRL**); [Baier *et al.* 2000]

- **automata-based**: properties are expressed in terms of an automaton
  - Continuous Stochastic Logic - Timed Automata (**CSL^{TA}**); [Haddad *et al.* 2007]
  - CTMC model checking against Deterministic Timed Automata; [Katoen *et al.* 2009]

Stochastic Logic

basic idea: to extend reasoning of classical temporal logic (LTL/CTL) to stochastic models models

- **syntax based**: properties are expressed in terms of a formula
  - Continuous Stochastic Logic (**CSL**); [Aziz 2000]
  - action-state Continuous Stochastic Logic (**asCSL**); [Baier *et al.* 2004]
  - Continuous Stochastic Reward Logic (**CSRL**); [Baier *et al.* 2000]

- **automata-based**: properties are expressed in terms of an automaton
  - Continuous Stochastic Logic - Timed Automata (**CSL^{TA}**); [Haddad *et al.* 2007]
  - CTMC model checking against Deterministic Timed Automata; [Katoen *et al.* 2009]

- **important points about above logics**:
  - limited to CTMCs models
  - designed for application of numerical methods for CTMC analysis
  - solution algorithms are affected by state-space explosion

Continuous Stochastic Logic

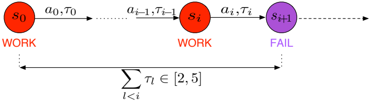
CSL syntax

$$\phi := a \mid tt \mid \neg\phi \mid \phi \wedge \phi \mid \mathcal{S}_{\sim p}(\phi) \mid \mathcal{P}_{\sim p}(\varphi) \qquad \text{(state-formulae)}$$

$$\varphi := X^I \phi \mid \phi\, U^I \phi \qquad \text{(path-formulae)}$$

$a \in AP$ (Atomic Propositions), $I = [t_1, t_2] \subseteq \mathbb{R}_{\geq 0}$, $\sim \in \{\leq, <, >, \geq\}$ and $p \in [0, 1]$

- $\sigma \models (\phi\, U^I \psi) \Leftrightarrow$ if a FUTURE state ($\sigma[i]$) satisfying ψ is reached within I through a sequence of ϕ-states



$\models$ (work $U^{[2,5]}$ fail)

- in general a logic formula φ shall allow to reason about any/all of the following:
  - state labels: $L(s_i)$
  - transition labels (actions): a_i
  - transition durations: τ_i
  - state/transition rewards (if supported)

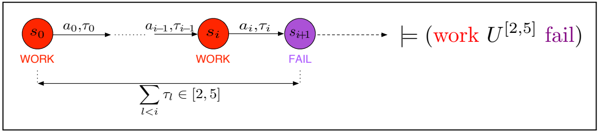
Stochastic Temporal reasoning

what type of properties can we characterize through stochastic temporal logics ?

- **reachability** (CSL)
- **sequential reachability** (asCSL)
- **multiple-bounded sequential reachability** (single-clock automata - CSL^{TA})
- **conjunction of multiple-bounded sequential reachability** (multiple-clock automata DTA-CSL)

Stochastic Temporal reasoning

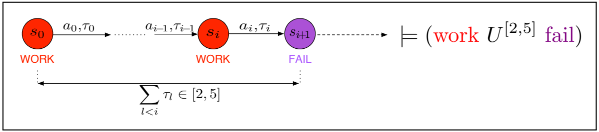
- **reachability [CSL]:** reasoning about state-labels + single-time-bound



$\Rightarrow$ *"reaching a fail state at time point* $t \in [2,5]$ *remaining in work-states until that point"*

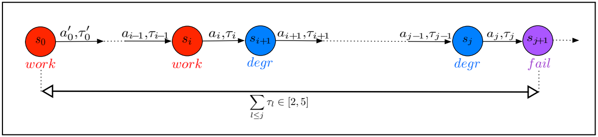
Stochastic Temporal reasoning

- **reachability [CSL]:** reasoning about state-labels + single-time-bound



⇒ *"reaching a fail state at time point* $t \in [2, 5]$ *remaining in work-states until that point"*

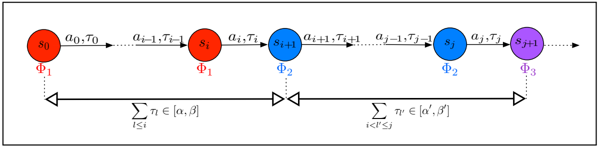
- **sequential reachability [asCSL]**: reasoning about state/action labels + single-time-bound



⇒ *"reach a fail-state within 2 and 5 time units passing through a sequence of work-states followed by degr-states"*

Stochastic Temporal reasoning

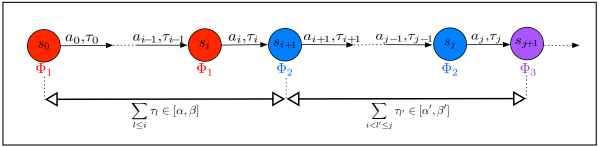
- **multiple-bounded sequential reachability [CSL^{TA}]**: reasoning about state/action labels + multiple-time-bounds



$\Rightarrow$ *"pass from work states to degr within [0,5] time units and then from degr to Fail states within [3,9] time units"*

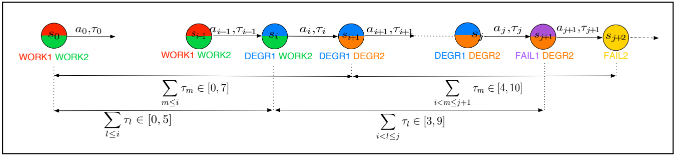
Stochastic Temporal reasoning

- **multiple-bounded sequential reachability [CSL^{TA}]**: reasoning about state/action labels + multiple-time-bounds



⇒ *"pass from work states to degr within [0,5] time units and then from degr to Fail states within [3,9] time units"*

- **conjunction of multiple-bounded sequential reachability [CSL^{DTA}]**: reasoning about state/action labels + independent-time-bounds



Φ : *"process P_1 will* **FAIL** *going from WORK1-to-DEGR1 states within 5 time units and then from DEGR1-to-FAIL1 within $[3, 9]$ time units* **AND** *process P_2 will* **FAIL** *going from WORK2-to-DEGR2 states within 7 time units and then from DEGR2-to-FAIL2 within $[4, 10]$ time units"*

Reward-based reasoning

- **Rewards**: real-valued functions of a model state (state-rewards) or transition (transition-reward)

- **reward-based reasoning**: temporal-reasoning + conditions on the reward accumulated

- **CSRL [Baier *et al.* 2000]**: CTMC + state-reward structure ($\rho : S \rightarrow \mathbb{R}^+$)
  - CSL path-operators with time-bounds + reward-bounds
    - $(\Phi \; U_J^I \Psi)$: *reach a Ψ-state (through Φ-states) within $t \in I$ and so that average reward cumulated spending time in Ψ-states is in J.*
  - $\rightarrow$ reward-analysis based on CTMC transient-analysis and steady-state analysis

Taxonomy of stochastic logics

	CSL	CSRL	asCSL	CSL^{TA}	DTA-CSL
formalism	syntax	syntax	reg. expr. on action/state	1-clock timed automata	N-clock timed automata
reachability	YES	YES	YES	YES	YES
reward-bounded reachability	NO	YES	NO	NO	NO
sequential reachability	NO	NO	YES	YES	YES
multiple-bounded sequential reachability	NO	NO	NO	YES	YES
nested-UNTIL and conjunction of multiple sequential reachability	NO	NO	NO	NO	YES
rewards	NO	YES	NO	NO	NO
type of model	CTMC	CTMC + state reward	CTMC	CTMC	CTMC
numerical solution	YES	YES	YES	YES	YES
statistical solution	YES[1]	YES	N/A	N/A	N/A

[1] only on sub-logic with no-nested path-operators

Outline

Stochastic Logics

CSL^{LHA}

Software support for CSL^{LHA}

CSL^{LHA}: going beyond probabilistic verification

- with existing Probabilistic Model Checking:
  - (only) evaluation of the probability of (possibly reward-bounded[2]) conditions
  - CTMC models
  - mostly Numerical (i.e. state-space explosion problem)

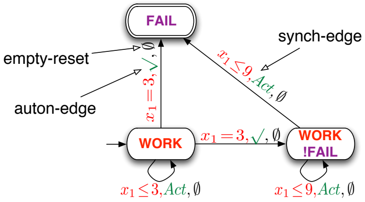
- with CSL^{LHA}
  - evaluation of the conditional expectation of “sophisticated" random variables (including Binomial ones)
  - any type of Discrete Event Stochastic Process
  - no state-space explosion problem (simulation based verification)

- how do we achieve that? through Linear Hybrid Automata

---

[2] CSRL

Deterministic Timed Automata



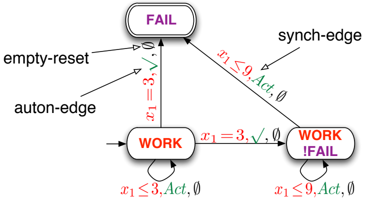
$\Longleftrightarrow$ (WORK $U^{[3,9]}$ FAIL)

- DTA: a machine that reads in paths $\sigma = s_0 \stackrel{a_0, \tau_0}{\longrightarrow} s_1 \stackrel{a_0, \tau_0}{\longrightarrow} \ldots$
- locations labeled with state formula ; n-tuple $X = (x_1, \ldots, x_n)$ of *clock variables*
- transitions: $l \stackrel{\gamma, A, r}{\longrightarrow} l'$
  - γ : a clocks' constraint e.g. $\bigwedge_i x_i \leq c_i$
  - A : a set of actions or $\surd$
  - a (possibly empty) set of clock resets

autonomous edge $\Leftrightarrow A = \surd$ synchronizing edge $\Leftrightarrow A \neq \surd$

auton-edges have precedence on synch-edges

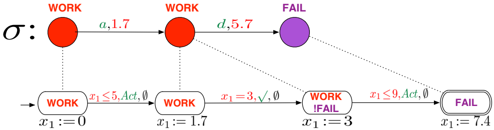
Deterministic Timed Automata



$\Longleftrightarrow$ (WORK $U^{[3,9]}$ FAIL)

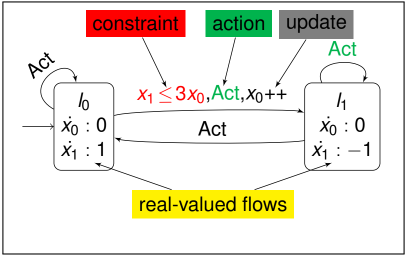
- reading of a path through synchronization with the automaton
  - path σ is accepted if it leads to a final location of the automata
  - path σ is rejected if synchronization blocks without reaching a final location

because of the Determinism of the automata, reading of a path is guaranteed to terminate



a DESP-transition corresponds to several DTA-auton-transitions preceeded/followed by the corresponding DTA-synch-transition

From DTA to Linear Hybrid Automata

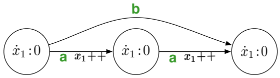


- LHA: a generalization of DTA
- n-tuple $X = (x_1, \ldots, x_n)$ of real-valued *data variables*
- a variable's flow:
  - depends on the location
  - can be a real-valued constant or a real-valued function of a DESP state
- transitions: $l \xrightarrow{\gamma, A, r} l'$
  - γ : a constraint as linear function of variables' value $\bigwedge_j \left(\sum_{1 \leq i \leq n} \alpha_i x_i \sim c \right)$

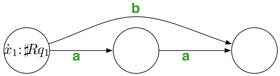
    $\sim \in \{<, \leq, \geq, >\}, =$
  - u : variable update (linear function) $u_k(X) = \sum_{1 \leq i \leq n} \alpha_i x_i + c$

Modelling with LHA variables

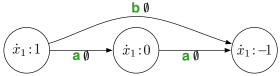
LHA variables can be used to model several things including: timers, state-rewards, transition-rewards



- x_1 : transition-reward
- zero-flow in every location
- update: function of synch-action (increment if event-counter)



- x_1 : state-reward
- flow: function of a DESP-state-indicator

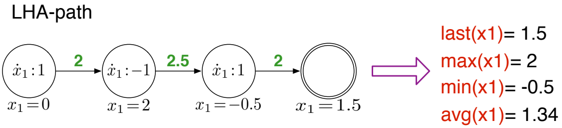


- x_1 : timers
- flow: normally in $\{1, 0, -1\}$
- no update

LHA Path-Variables

- a path-variable Y is a random variable corresponding to a synchronizing-path of an LHA

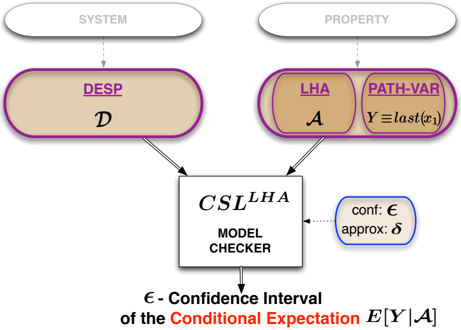
  - $Y \equiv last(x_i)$ *the last value of* x_i *along the synchronizing path*
  - $Y \equiv min(x_i)$ *the minimal value of* x_i *along the synchronizing path*
  - $Y \equiv avg(x_i)$ *the average value of* x_i *along the synchronizing path*
  - $Y \equiv var(x_i)$ *the variability of the value of* x *along the synchronizing path*
  - $Y \equiv corel(x_i, x_j)$ the correlation between the value of x_i and x_j along the synchronizing path.



- path-variables are evaluated *on-the-fly* on generation of a $(\mathcal{D} \times \mathcal{A})$-path

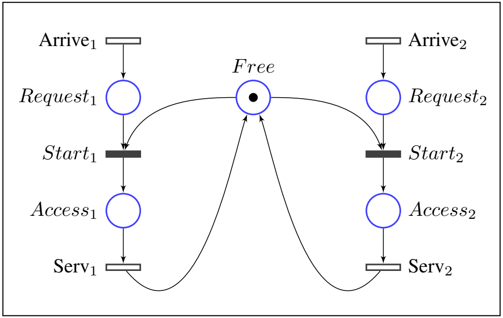
CSL^{LHA} model checking

- **input**: an LHA + a Path-Variable Y
  - **model**: a DESP
  - **formula**: an LHA + a Path-Variable Y
  - confidence-level: ϵ; approximation (interval-width): δ;
- **output**: confidence interval of the conditional-expectation $E[Y|\mathcal{A}]$



Running example

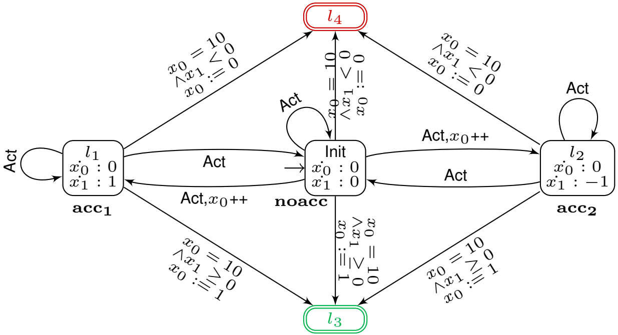
example: A shared resource model



- infinite-state DESP
- Arrivals$\sim Exp()$; Services$\sim Unif()$
- immediate-transitions ($Start_1$,$Start_2$) have weights w_1 and w_2 and equal priorities

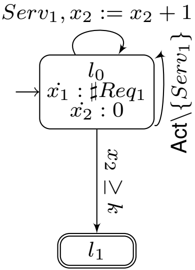
Example 1: utilization difference measures

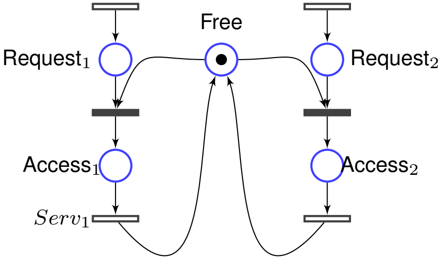
after 10 times the resource has been used, P_1-processes have used longer the resource than P_2's



- x_0 (overloaded variable): service counter (transition reward) before acception; bernoulli variable (on acceptance)
- x_1 : timer, difference between occupation time by P_1 and P_2
- $last(x_0)$: probability that the shared resource is used longer by P_1-processes than by P_2's.
- $last(x_1)$: difference between the utilization time of P_1 and P_2 processes
- $avg(x_1)$: (*resp.* $min(x_1)$) average value (*resp.* minimum) of such difference

Example 2: average waiting time til k departures





- x_1 : P_1-processes cumulative waiting time (state-reward variable)
- x_2 : number of P_1-processes that have used the resource (transition-reward variable)
- $last(x_1/x_2)$: (Sup of the) average waiting time (until k P_1-departures)

Taxonomy of stochastic logics

	CSL	CSRL	asCSL	CSL^{TA}	DTA-CSL	CSL^{LHA}
formalism	syntax	syntax	reg. expr. on action/state	1-clock DTA	N-clocks DTA	LHA
reachability	YES	YES	YES	YES	YES	YES
reward-bounded reachability	NO	YES	NO	NO	NO	YES
sequential reachability	NO	NO	YES	YES	YES	YES
multiple-bounded sequential reachability	NO	NO	NO	YES	YES	YES
nested-UNTIL and conjunction of multiple sequential reachability	NO	NO	NO	NO	YES	YES
rewards	NO	state-only	NO	NO	NO	YES
type of model	CTMC	CTMC + state reward	CTMC	CTMC	CTMC	DESP
numerical solution	YES	YES	YES	YES	YES	NO
statistical solution	YES[3]	YES	N/A	N/A	N/A	YES

[3] only on sub-logic with no-nested path-operators

Outline

Stochastic Logics

CSL^{LHA}

Software support for CSL^{LHA}

COSMOS: a software tool for CSL^{LHA} analysis

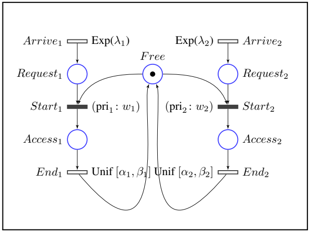
- implemented in C++

- inputs:
  - a DESP expressed as a GSPN (with generally distributed timed transitions)
  - an LHA + a path-variable Y + confidence level (ϵ) + approximation level (δ)

- output: iterative computation of confidence interval estimation of the $E[Y|\mathcal{A}]$

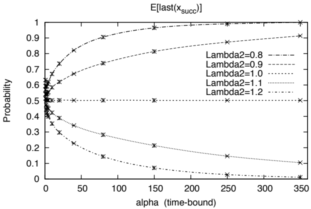
Some experiments with COSMOS

assessing resource occupation based measures as a function of the arrival-rate λ_2 (arrival of P_2-processes)

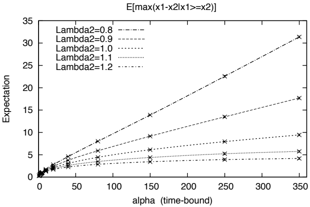


time bound $I = [\alpha, \alpha]$,
x_0: real-time;
x_1: occupation divergence timer;
acceptance condition: $x_1 > 0) \wedge (x_0 \in [\alpha, 2\alpha])$

- $\Phi_1 \equiv E\left[success|(x_1 > 0) \wedge (x_0 \in [\alpha, \alpha])\right]$
  probability that Res used longer by P_1-processes
- $\Phi_2 \equiv E\left[max(x_1)|(x_1 > 0) \wedge (x_0 \in [\alpha, \alpha])\right]$
  maximum occupation time divergence
- $\Phi_3 \equiv E\left[avg(x_1)|(x_1 > 0) \wedge (x_0 \in [\alpha, \alpha])\right]$
  average occupation time divergence



(a) Φ_1: probability of $x_1 \geq x_2$



(b) Φ_2: maximum positive difference $max(x_1 - x_2)$

Conclusion

CSL^{LHA} a logic for stating complex temporal properties of stochastic models

- it unifies expressive temporal reasoning with reward based analysis (concept of conditional expectation)
- it naturally leans towards a Perfomance Evaluation model checking approach
- it targets DESP models (not only Markovian)
- it uses a statistical (simulation based) approach to estimate measure of interest
- it does not suffer of state-space-explosion
- software support available (prototype)

Future developments

- tool: code-optimization (on-going); GUI development (yet to start)
- optimization of the simulation process wrt rare-events
- ...

THANK YOU FOR YOUR ATTENTION

REFERENCE: TechReport=TR-LACL-2010-8,
author = P. Ballarini, H. Djafri, M. Duflot, S. Haddad, N. Pekergin.
title = CSL^{LHA}: an Expressive Language for Statistical Verification of Stochastic Models.
institution = LACL (Laboratory of Algorithms, Complexity and Logic), University of Paris-Est (UPEC),
year = 2010, number = TR-LACL-2010-8,
PDF = http://lacl.u-pec.fr/Rapports/TR/TR-LACL-2010-8.pdf

acknoweldgements: this work has been partially funded by the ANR Checkbound project