A symbolic calculus on Symmetric Net arc functions: applications

Joint work with: Lorenzo Capra, Massimiliano De Pierro, Marco Beccuti, Simone Pernice

Stochastic Symmetric Nets

- A Colored Petri Net-like formalism proposed in the ’90s
- A «color syntax» devised to exploit (most) model symmetries during analysis
  - State space methods can benefit: lumpabiliy is exploited, (much) smaller state spaces are generated directly
  - Structural properties derivation (interesting features of PNs): define a language of expressions similar to arc functions and operators on them
  - The language for expressing structural properties is useful also for the generation of a set of *Symbolic* Ordinary Differential Equations to derive expected values of performance indices of interest.

EPEW 2018

Deriving Symbolic ODE from SSNs without unfolding

M. Beccuti, L. Capra, M. De Pierro,
G. Franceschinis, S. Pernice

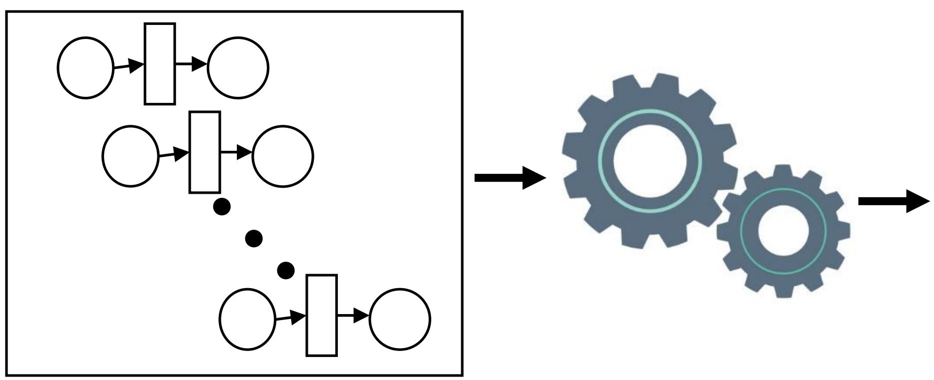
Università di Torino

Università di Milano

Università del Piemonte Orientale

Fluidification of Stochastic Petri Nets

Analysis methods that approximate the behavior of the stochastic process underlying an SPN with a deterministic one, modelled through a system of Ordinary Differential Eqs. (under given assumptions [2][9]). Expected value at time T of state components can be obtained by solving the system of ODE.



$dx_1/dt = +\lambda x_2 - \mu\, x_3$

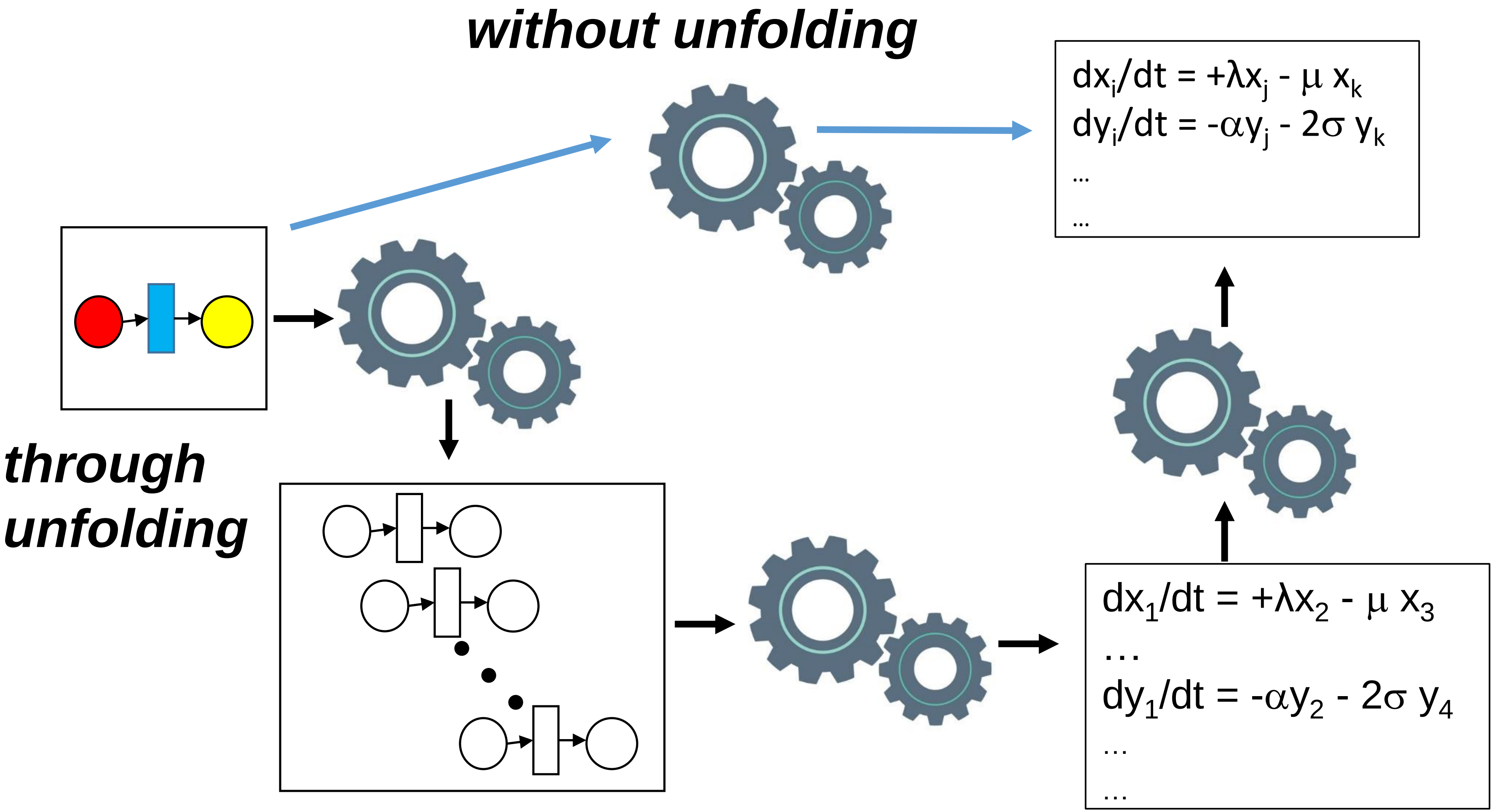
…

$dy_1/dt = -\alpha y_2 - 2\sigma\, y_4$

…

…

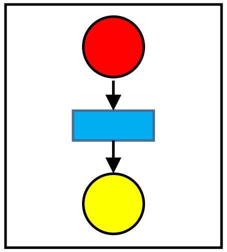
Fluidification of Stocastic Symmetric Nets



Outline

- Stochastic Symmetric Nets
- Ordinary Differential Equations vs Symbolic ODE
- Direct generation of SODE:
  - SSN partial unfolding
  - Symbolic ODE terms: transposing arc functions
  - Evaluation of the enabling degree of symbolic instances
  - Cardinality of symbolic expressions
- Tool support
- Concluding remarks
- Related work and Future work

Stochastic Symmetric Nets



Petri nets with colored tokens and transition instances + stochastic (exponentially distributed) firing times

- Compact representation of models
- Structured syntax of colors
  - Lumped State Space and Stochastic Process (CTMC)
  - Symbolic computation of structural properties
- Unfolding of an SSN model into a SPN enables the application of analysis methods defined only for SPNs (e.g. fluid approximation).

Color domain

$Mac \times Loc$

Arc functions: I(t,p), O(t,p)

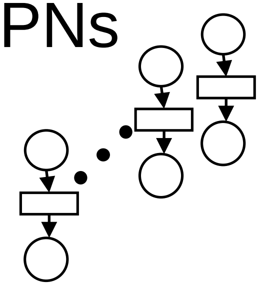
$\langle x, l\rangle[x \in N]$

$\langle l, l'\rangle[l \neq l']$

Expressions (arc fn. –like) and Operators

$[c_3 \neq c_2]\langle S - c \cap S_{C1}, c, c'\rangle[c \in C1]$

$(f_1 - f_2)^t \bullet f_3$



Exploiting symmetries

Groups of places in the unfolded model have «similar» ODE $\Rightarrow$ under the hypothesis of SYMMETRIC INITIAL MARKING the expected number of tokens in these places remains uniform in time [4].

Only one representative in each «equivalence class» is used to derive the solution. A Symbolic ODE is an ODE representing the whole class.

[4] M. Beccuti, C. Fornari, G. Franceschinis, S. Halawani, O. Ba-Rukab, A. Ahmad, and G. Balbo. From symmetric nets to differential equations exploiting model symmetries. Computer Journal, 58(1):23-39, 2015.

Exploiting symmetries

$$\begin{aligned}
\frac{dx_{P_{1,\alpha}}}{dv} &= -\lambda(\min[x_{P_{1,\alpha}}, x_{P_{2,\beta}}]) - \lambda(\min[x_{P_{1,\alpha}}, x_{P_{2,\delta}}]) \\
\frac{dx_{P_{1,\beta}}}{dv} &= -\lambda(\min[x_{P_{1,\beta}}, x_{P_{2,\alpha}}]) - \lambda(\min[x_{P_{1,\beta}}, x_{P_{2,\delta}}]) \\
\frac{dx_{P_{1,\delta}}}{dv} &= -\lambda(\min[x_{P_{1,\delta}}, x_{P_{2,\alpha}}]) - \lambda(\min[x_{P_{1,\delta}}, x_{P_{2,\beta}}])
\end{aligned}$$

$$\begin{aligned}
\frac{dx_{P_{2,\alpha}}}{dv} &= -\lambda(\min[x_{P_{1,\beta}}, x_{P_{2,\alpha}}]) - \lambda(\min[x_{P_{1,\delta}}, x_{P_{2,\alpha}}]) \\
\frac{dx_{P_{2,\beta}}}{dv} &= -\lambda(\min[x_{P_{1,\alpha}}, x_{P_{2,\beta}}]) - \lambda(\min[x_{P_{1,\delta}}, x_{P_{2,\beta}}]) \\
\frac{dx_{P_{2,\delta}}}{dv} &= -\lambda(\min[x_{P_{1,\alpha}}, x_{P_{2,\delta}}]) - \lambda(\min[x_{P_{1,\beta}}, x_{P_{2,\delta}}])
\end{aligned}$$

$$\begin{aligned}
\frac{dx_{P_{3,\alpha,\beta}}}{dv} &= +\lambda(\min[x_{P_{1,\alpha}}, x_{P_{2,\beta}}]) \\
\frac{dx_{P_{3,\alpha,\delta}}}{dv} &= +\lambda(\min[x_{P_{1,\alpha}}, x_{P_{2,\delta}}]) \\
\frac{dx_{P_{3,\beta,\alpha}}}{dv} &= +\lambda(\min[x_{P_{1,\beta}}, x_{P_{2,\alpha}}]) \\
\frac{dx_{P_{3,\beta,\delta}}}{dv} &= +\lambda(\min[x_{P_{1,\beta}}, x_{P_{2,\delta}}]) \\
\frac{dx_{P_{3,\delta,\alpha}}}{dv} &= +\lambda(\min[x_{P_{1,\delta}}, x_{P_{2,\alpha}}]) \\
\frac{dx_{P_{3,\delta,\beta}}}{dv} &= +\lambda(\min[x_{P_{1,\delta}}, x_{P_{2,\beta}}])
\end{aligned}$$

$$\begin{aligned}
\frac{d\widehat{x}_{P_{1,Z^1_{1,1}}}}{dv} &= -\lambda(\min(\widehat{x}_{P_{1,Z^1_{1,1}}}, \widehat{x}_{P_{2,Z^1_{1,1}}}) - \lambda(\min(\widehat{x}_{P_{1,Z^1_{1,1}}}, \widehat{x}_{P_{2,Z^1_{1,1}}}) \\
\frac{d\widehat{x}_{P_{2,Z^1_{1,1}}}}{dv} &= -\lambda(\min(\widehat{x}_{P_{1,Z^1_{1,1}}}, \widehat{x}_{P_{2,Z^1_{1,1}}}) - \lambda(\min(\widehat{x}_{P_{1,Z^1_{1,1}}}, \widehat{x}_{P_{2,Z^1_{1,1}}}) \\
\frac{d\widehat{x}_{P_{3,Z^1_{1,1},Z^2_{1,1}}}}{dv} &= +\lambda(\min(\widehat{x}_{P_{1,Z^1_{1,1}}}, \widehat{x}_{P_{2,Z^1_{1,1}}})
\end{aligned}$$

[4] M. Beccuti, C. Fornari, G. Franceschinis, S. Halawani, O. Ba-Rukab, A. Ahmad, and G. Balbo. From symmetric nets to differential equations exploiting model symmetries. Computer Journal, 58(1):23-39, 2015.

Goal: efficient computation of measures

Exploit the regular structure of SSN models: use only a subset of ODEs, representing all the others, to derive the expected value of the interesting measures

Symbolic ODE

Exploit the calculus developed for structural properties computation of SSNs[5][6][7] to avoid unfolding and reduction of ODE system

Exploiting the calculus on arc expressions

The SODE for (unfolded) place *P[c]* contains a negative/positive term for each *transition instance* that causes a decrease/increase of c-colored tokens in *P*.

The SSN arc functions express in a *symbolic form* the state change that each transition instance causes on the colored tokens in its input and output places.

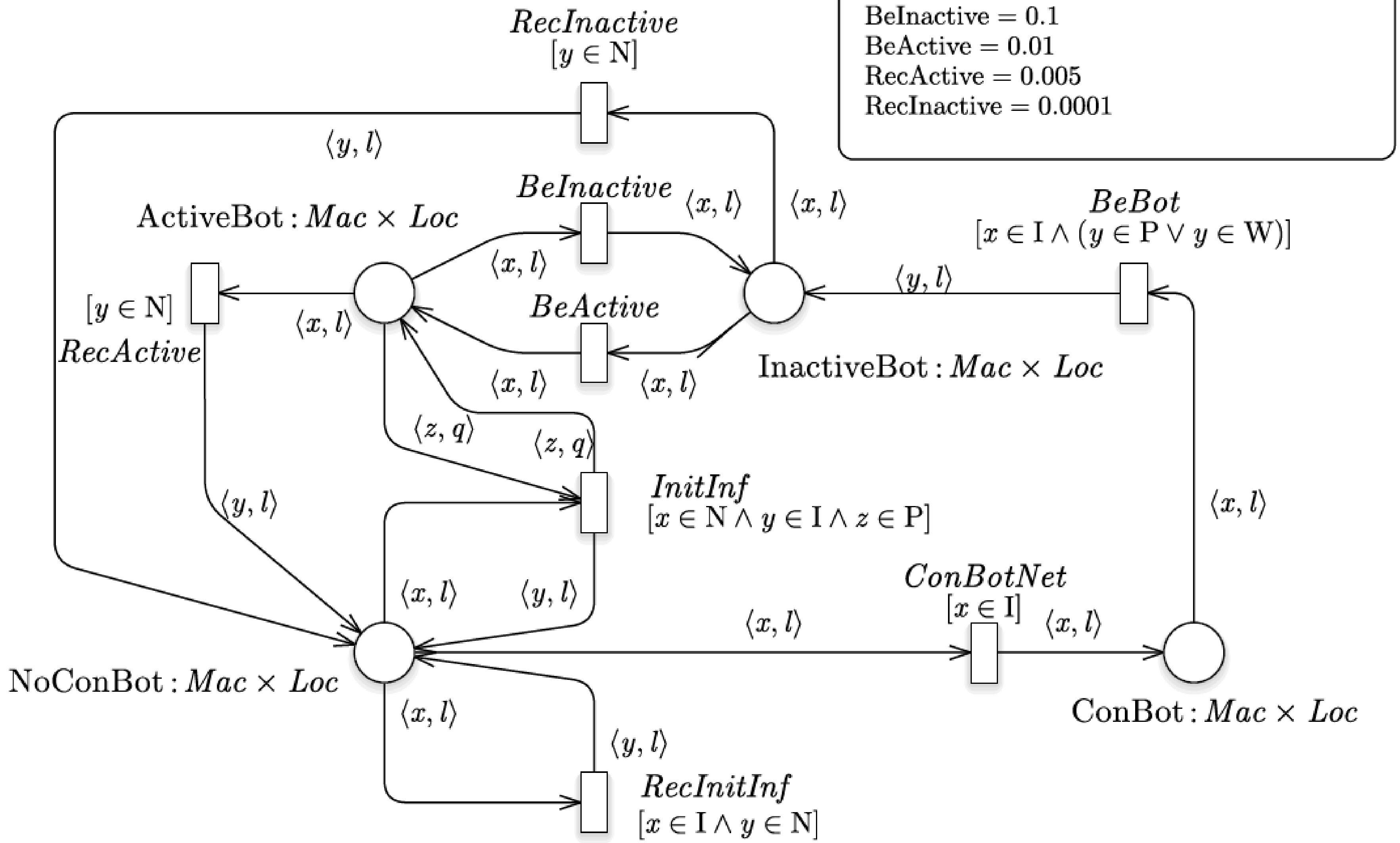
IDEA: This symbolic form can be used to find the (set of) transition instance(s) changing the number of c-colored tokens in *P*

class $Mac = n\{1..1\}$ **is** N $+\ i\{1..1\}$ **is** I $+\ q\{1..1\}$ **is** P $+\ w\{1..1\}$ **is** W
class $Loc = l\{1..10\}$

$$m_0 = \mathrm{NoConBot}\big(1000\,\langle \mathrm{N}, \boldsymbol{Loc}\rangle\big) + \mathrm{ConBot}\big(\langle \mathrm{I}, \boldsymbol{Loc}\rangle\big)$$

Rates

InitInf = same loc. 10; different loc. 2
BeBot = mach $\in$ W : 20, mach $\in$ P : 2
RecInitInf = 0.5
ConBotNet = 3
BeInactive = 0.1
BeActive = 0.01
RecActive = 0.005
RecInactive = 0.0001



Example inspired by [12]

Ordinary Differential Equations

Based on the UNFOLDED MODEL (SPN)

Two color classes: Mac = Machine states (N, I, P, W)

Loc = Local Area Network id

Size of the unfolded model:

- Places: $4 \cdot |Mac| \cdot |Loc| = 16 \cdot |Loc|$ (but $5 \cdot |Loc|$ isolated and empty)
- Transitions: $|Loc|$ (ConBot) + $2 \cdot |Loc|$ (BeBot) + $4 \cdot |Loc|$ (BeActive) + $4 \cdot |Loc|$ (BeInactive) + $4 \cdot |Loc|$ (RecActive) + $4 \cdot |Loc|$ (RecInactive) + $|Loc|$ (RecInitInf) + $|Loc|^2$ (InitInf) $= 20 \cdot |Loc| + |Loc|^2$

Size of the ODE system of the unfolded model:
$11|Loc|$ equations and $2 \cdot (20|Loc|+|Loc|^2)$ terms

ODE vs Symbolic ODE

|Loc|	Terms ODE	ODE/SODE	Mean Solution time(sec) ODE	SODE
1	42 (11 eq)	2.51 (1.57)	0.37	0.085
10	600 (110 eq.)	21.43 (15.7)	38.81	0.238
20	1600 (220 eq.)	57.14 (31.43)	572.18	0.292
50	7000 (550 eq)	250 (78.58)	>4h	0.248

Size of the ODE system of the unfolded model: $11 \cdot |Loc|$ equations and $2 \cdot (20 \cdot |Loc| + |Loc|^2)$ terms

Size of the SODE system for the same model: 7 SODE (+4 always = 0) with 28 terms: <u>independent of |Loc|</u>

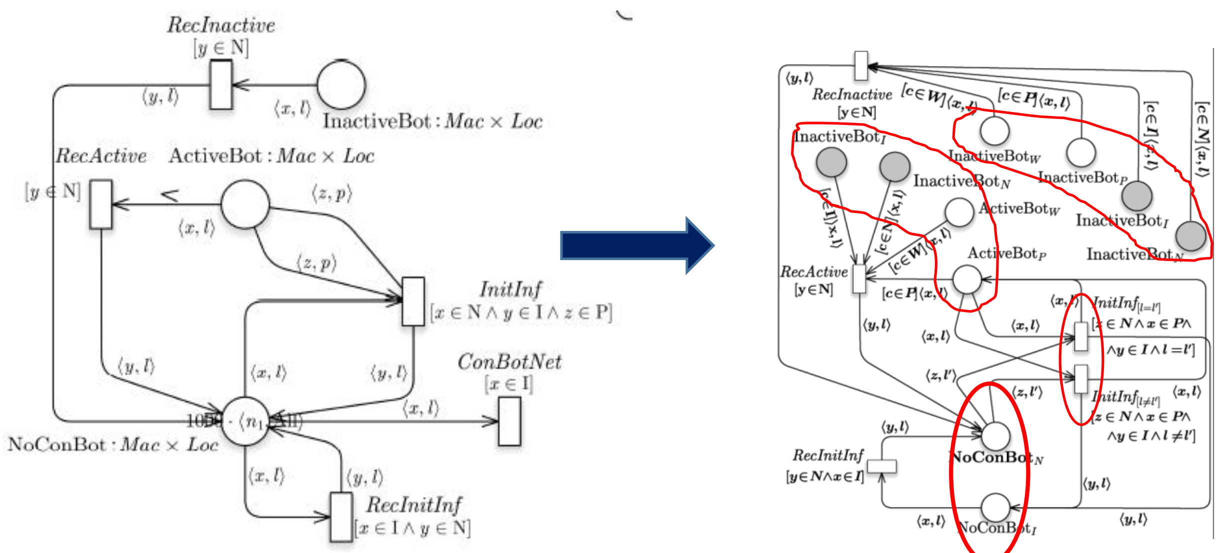
For large |Loc| the unfolded SPN grows very large and the approach becomes impractical

Steps to derive the system of SODE

- Partial unfolding of the places and transitions
- Derivation of «macro terms» in the symbolic ODE: functions Added-By and Removed-By, AB and RB
- Computation of the base rate and of the enabling degree of the terms: possible refinement of terms in subterms with homogeneous enabling degree
- Computation of the cardinality of the sets of transition instances represented by the refined terms (producing the same state changes with same rate)

Partial unfolding

- Partition of color classes into (static) subclasses
- Partition of place color domains based on color subclasses and comparison of color elements of same type ⇒ place partial unfolding
- Partition of transition instances based on their rate ⇒ transition partial unfolding



Sum of terms in the SODE

Added-By and Removed-By: transposing arc functions

$$\frac{dx[p,c]}{d\nu} = \sum_{\langle t,c'\rangle : p\in t^{\bullet}, c'\in \boxed{\mathcal{A}(t,p)(c)}} \varphi(x(\nu),t,c')\left(\overbrace{\boxed{\mathcal{A}(t,p)(c)[c']}}^{\text{Number of c-colored tokens added by } \langle t, c' \rangle \text{ in } p}\right) - \sum_{\langle t,c'\rangle : p\in {}^{\bullet}t, c'\in \boxed{\mathcal{R}(t,p)(c)}} \varphi(x(\nu),t,c')\left(\overbrace{\boxed{\mathcal{R}(t,p)(c)[c']}}^{\text{Number of c-colored tokens withdrawn by } \langle t, c' \rangle \text{ from } p}\right)$$

$$\left.\begin{aligned}
&\mathcal{A}(t,p) : cd(p) \to Bag[cd(t)];\ \mathcal{A}(t,p) = (O[t,p] - I[t,p])^{t} \\
&\mathcal{R}(t,p) : cd(p) \to Bag[cd(t)];\ \mathcal{R}(t,p) = (I[t,p] - O[t,p])^{t}
\end{aligned}\right\} \text{Expressed with the same syntax of arc functions}$$

place $(NoConBot_N, c, l)$;$g = c \in N$	
$\mathcal{A}(RecActive, .)$	$\langle S_{Mac}, c, l\rangle[g]$
$\mathcal{A}(RecInactive, .)$	$\langle S_{Mac}, c, l\rangle[g]$
$\mathcal{R}(InitInf_{[l=q]}, .)$	$\langle c, S_I, S_P, l, l\rangle[g]$
$\mathcal{R}(InitInf_{[l\neq q]}, .)$	$\langle c, S_I, S_P, l, S-l\rangle[g]$
$\mathcal{A}(RecInitInf, .)$	$\langle S_I, c_1, l\rangle[g]$

The terms may group several instances that add/withdraw the same number of tokens (e.g. $|Loc| - 1$ instances withdraw 1 token)

Enabling degree computation

- Composition of I/O arc fns with representative instance

$$\frac{dx[p,c]}{d\nu} = \sum_{\langle t,c'\rangle : p \in t^{\bullet}, c' \in \mathcal{A}(t,p)(c)} \boxed{\varphi(x(\nu),t,c')}(\mathcal{A}(t,p)(c)[c']) - \sum_{\langle t,c'\rangle : p \in {}^{\bullet}t, c' \in \mathcal{R}(t,p)(c)} \boxed{\varphi(x(\nu),t,c')}(\mathcal{R}(t,p)(c)[c']).$$

$$\varphi(x(\nu),t_j) = \omega(t_j) \min_{l:I[p_l,t_j]\neq 0} \frac{x_l(\nu)}{I[p_l,t_j]}$$

Intensity of t_j (where t_j corresponds to a transition instance $\langle t, c' \rangle$

The enabling degree of instance $\langle t, c' \rangle$ must be evaluated for the instances grouped in a term of $A(t,p)(c)$ or $R(t,p)(c)$ and *may not be uniform*

Partial unfolding ensures uniform rate $\omega(t_j)$ for all instances $\langle t, c' \rangle$ of any given t

Composition of Input arc fn. with a representative transition color instance provides the denominator; it may refine the set of instances into sub-sets (by rewriting the term as a sum of disjoint terms).

Cardinality of terms in $A(t,p)$ and $\mathrm{R}(t,p)$

$A(t,p)$ and $\mathrm{R}(t,p)$: sums of constant size terms (Property 2)

$$\frac{dx[p,c]}{d\nu} = \sum_{t:p\in t^{\bullet}, \boxed{F_i\ in\ \mathcal{A}(t,p)}} \lambda_i n_i \varphi(x(\nu),t) - \sum_{t:p\in {}^{\bullet}t, \boxed{F_j\ in\ \mathcal{R}(t,p)}} \lambda_j n_j \varphi(x(\nu),t)$$

$\boldsymbol{n_i = |F_i|, n_j = |F_j|}$

λ_i / λ_j multiplicity of $\boldsymbol{F_i/F_j}$ in $A(t,p)$ / $\mathrm{R}(t,p)$

For each term F_i representing several transition instances with uniform intensity φ, only one term is generated in the SODE and multiplied by $|F_i|$.

$$\frac{dx[NoConBot_N,c,l]}{d\nu} = |P|\omega_1 x[ActiveBot_P,c',l] + |W|\omega_2 x[ActiveBot_W,c',l] +$$
$$+ |P|\omega_3 x[InactiveBot_P,c',l] + |W|\omega_4 x[InactiveBot_W,c',l] + \omega_7 x[NoConBot_I,c',l] +$$
$$- |P||I|\omega_5 \min(x[NoConBot_I,c',l], x[ActiveBot_P,c'',l]) +$$
$$- |P||I|(|Loc|-1)\omega_6 \min(x[NoConBot_I,c',l], x[ActiveBot_P,c'',l]),$$

Tool support: www.di.unito.it/~depierro/SNex

- Library implementing algorithms for the computation of transpose, difference and a (restricted) composition operator. It is based on a set of rewriting rules.
- Command Line Interface (CLI)
  - CLI commands for – loading a SSN specification, «solving» an expression involving the above operators.
  - CLI commands to operate all the intermediate steps required to finally derive the Symbolic ODEs

Concluding remarks

- Implementation in SNexpression (to be completed)
- Second example in the Tech Report [3]
- The method works also for ordered classes

Related work and Future work

- Investigate the applicability to other formalism with similar symmetric structure [14](call for collaboration)
- Complete the implementation and extensive experimentation on large models
- Automatic precondition check of applicability exploiting the same symbolic calculus

[14] M. Tschaikowski and M. Tribastone. Exact fluid lumpability for Markovian process algebra. In M. Koutny and I. Ulidowski, editors, CONCUR 2012 pages 380-394. Springer Berlin Heidelberg, 2012.