Performance evaluation of the shared memory system in $dtsPBC$

Igor V. Tarasyuk

A.P. Ershov Institute of Informatics Systems

Siberian Division of the Russian Academy of Sciences

6, Acad. Lavrentiev pr., Novosibirsk 630090, Russia

itar@iis.nsk.su

db.iis.nsk.su/persons/itar

Abstract: Algebra $dtsPBC$ is a discrete time stochastic extension of finite Petri box calculus (PBC) enriched with iteration.

In this work, within dtsPBC, a method of modeling and performance evaluation based on stationary behaviour analysis for concurrent systems is outlined applied to the shared memory system.

Keywords: stochastic process algebra, Petri box calculus, discrete time, iteration, stationary behaviour, performance evaluation, shared memory system.

Contents

Introduction

Algebra PBC and its extensions

- *Petri box calculus PBC* [BDH92]
- *Time Petri box calculus $tPBC$* [Kou00]
- *Timed Petri box calculus $TPBC$* [MF00]
- *Stochastic Petri box calculus $sPBC$* [MVF01,MVCC03]
- *Ambient Petri box calculus $APBC$* [FM03]
- *Arc time Petri box calculus $atPBC$* [Nia05]
- *Generalized stochastic Petri box calculus $gsPBC$* [MVCR08]
- *Discrete time stochastic Petri box calculus $dtsPBC$* [Tar05,Tar06]
- *Discrete time stochastic and immediate Petri box calculus $dtsiPBC$* [TMV10]

Syntax

The *set of all finite multisets* over X is $\mathbb{N}_f^X$.

The *set of all subsets* of X is 2^X.

$Act = \{a, b, \ldots\}$ is the set of *elementary actions*.

$\widehat{Act} = \{\hat{a}, \hat{b}, \ldots\}$ is the set of *conjugated actions (conjugates)* s.t. $a \neq \hat{a}$ and $\hat{\hat{a}} = a$.

$\mathcal{A} = Act \cup \widehat{Act}$ is the set of *all actions*.

$\mathcal{L} = \mathbb{N}_f^{\mathcal{A}}$ is the set of *all multiactions*.

The *alphabet* of $\alpha \in \mathcal{L}$ is $\mathcal{A}(\alpha) = \{x \in \mathcal{A} \mid \alpha(x) > 0\}$.

An *activity (stochastic multiaction)* is a pair (α, ρ), where
$\alpha \in \mathcal{L}$ and $\rho \in (0; 1)$ is the *probability* of multiaction α.

$\mathcal{SL}$ is the set of *all activities*.

The *alphabet* of $(\alpha, \rho) \in \mathcal{SL}$ is $\mathcal{A}(\alpha, \rho) = \mathcal{A}(\alpha)$.

The *alphabet* of $\Gamma \in \mathbb{N}_f^{\mathcal{SL}}$ is $\mathcal{A}(\Gamma) = \cup_{(\alpha,\rho)\in\Gamma}\mathcal{A}(\alpha)$.

For $(\alpha, \rho) \in \mathcal{SL}$, its *multiaction part* is $\mathcal{L}(\alpha, \rho) = \alpha$ and its *probability part* is $\Omega(\alpha, \rho) = \rho$.

The *multiaction part* of $\Gamma \in \mathbb{N}_f^{\mathcal{SL}}$ is $\mathcal{L}(\Gamma) = \sum_{(\alpha,\rho)\in\Gamma} \alpha$.

The operations: *sequential execution* ;, *choice* $[]$, *parallelism* $\|$, *relabeling* $[f]$, *restriction* rs, *synchronization* sy and *iteration* $[**]$.

Sequential execution and choice have the standard interpretation.

Parallelism does not include synchronization unlike that in standard process algebras.

Relabeling functions $f : \mathcal{A} \to \mathcal{A}$ are bijections preserving conjugates: $\forall x \in \mathcal{A}\ f(\hat{x}) = \widehat{f(x)}$.
For $\alpha \in \mathcal{L}$, let $f(\alpha) = \sum_{x\in\alpha} f(x)$. For $\Gamma \in I\!N_f^{\mathcal{SL}}$, let $f(\Gamma) = \sum_{(\alpha,\rho)\in\Gamma}(f(\alpha), \rho)$.

Restriction over an action a: any process behaviour containing a or its conjugate $\hat{a}$ is not allowed.

Let $\alpha, \beta \in \mathcal{L}$ be two multiactions s.t. for $a \in Act$ we have $a \in \alpha$ and $\hat{a} \in \beta$ or $\hat{a} \in \alpha$ and $a \in \beta$.
Synchronization of α and β by a is $\alpha \oplus_a \beta = \gamma$:

$$\gamma(x) = \begin{cases} \alpha(x) + \beta(x) - 1, & x = a \text{ or } x = \hat{a}; \\ \alpha(x) + \beta(x), & \text{otherwise}. \end{cases}$$

In the iteration, the initialization subprocess is executed first,
then the body one is performed zero or more times, finally, the termination one is executed.

Static expressions specify the structure of processes.

Definition 1 *Let* $(\alpha, \rho) \in \mathcal{SL}$ *and* $a \in Act$*. A* static expression *of* $dtsPBC$ *is*

$$E ::= \ (\alpha, \rho) \mid E;E \mid E[]E \mid E\|E \mid E[f] \mid E \text{ rs } a \mid E \text{ sy } a \mid [E*E*E].$$

$StatExpr$ is the set of *all static expressions* of $dtsPBC$.

Definition 2 *Let* $(\alpha, \rho) \in \mathcal{SL}$ *and* $a \in Act$*. A* regular static expression *of* $dtsPBC$ *is*

$$E ::= \ (\alpha, \rho) \mid E;E \mid E[]E \mid E\|E \mid E[f] \mid E \text{ rs } a \mid E \text{ sy } a \mid [E*D*E],$$

where $D ::= \ (\alpha, \rho) \mid D;E \mid D[]D \mid D[f] \mid D \text{ rs } a \mid D \text{ sy } a \mid [D*D*E].$

$RegStatExpr$ is the set of *all regular static expressions* of $dtsPBC$.

Dynamic expressions specify the states of processes.

Dynamic expressions are combined from static ones annotated with upper or lower bars.

The *underlying static expression* of a dynamic one: removing all upper and lower bars.

Definition 3 *Let $E \in StatExpr$ and $a \in Act$. A* dynamic expression *of $dtsPBC$ is*

$$G ::= \ \overline{E} \mid \underline{E} \mid G;E \mid E;G \mid G[]E \mid E[]G \mid G\|G \mid G[f] \mid G \text{ rs } a \mid G \text{ sy } a \mid$$
$$[G{*}E{*}E] \mid [E{*}G{*}E] \mid [E{*}E{*}G].$$

$DynExpr$ is the set of *all dynamic expressions* of $dtsPBC$.

A *regular dynamic expression*: its underlying static expression is regular.

$RegDynExpr$ is the set of *all regular dynamic expressions* of $dtsPBC$.

Operational semantics

Inaction rules

Inaction rules: instantaneous structural transformations.

Let $E, F, K \in RegStatExpr$ and $a \in Act$.

Inaction rules for overlined and underlined regular static expressions

$\overline{E;F} \Rightarrow \overline{E};F$	$\underline{E};F \Rightarrow E;\overline{F}$	$E;\underline{F} \Rightarrow \underline{E;F}$
$\overline{E[]F} \Rightarrow \overline{E}[]F$	$\overline{E[]F} \Rightarrow E[]\overline{F}$	$\underline{E}[]F \Rightarrow \underline{E[]F}$
$E[]\underline{F} \Rightarrow \underline{E[]F}$	$\overline{E \Vert F} \Rightarrow \overline{E} \Vert \overline{F}$	$\underline{E} \Vert \underline{F} \Rightarrow \underline{E \Vert F}$
$\overline{E[f]} \Rightarrow \overline{E}[f]$	$\underline{E}[f] \Rightarrow \underline{E[f]}$	$\overline{E \text{ rs } a} \Rightarrow \overline{E} \text{ rs } a$
$\underline{E} \text{ rs } a \Rightarrow \underline{E \text{ rs } a}$	$\overline{E \text{ sy } a} \Rightarrow \overline{E} \text{ sy } a$	$\underline{E} \text{ sy } a \Rightarrow \underline{E \text{ sy } a}$
$\overline{[E*F*K]} \Rightarrow [\overline{E}*F*K]$	$[\underline{E}*F*K] \Rightarrow [E*\overline{F}*K]$	$[E*\underline{F}*K] \Rightarrow [E*\overline{F}*K]$
$[E*\underline{F}*K] \Rightarrow [E*F*\overline{K}]$	$[E*F*\underline{K}] \Rightarrow \underline{[E*F*K]}$	

Let $E, F \in RegStatExpr,\ G, H, \widetilde{G}, \widetilde{H} \in RegDynExpr$ and $a \in Act$.

Inaction rules for arbitrary regular dynamic expressions

$$\frac{G \Rightarrow \widetilde{G},\ \circ \in \{;, []\}}{G \circ E \Rightarrow \widetilde{G} \circ E} \qquad \frac{G \Rightarrow \widetilde{G},\ \circ \in \{;, []\}}{E \circ G \Rightarrow E \circ \widetilde{G}} \qquad \frac{G \Rightarrow \widetilde{G}}{G \| H \Rightarrow \widetilde{G} \| H} \qquad \frac{H \Rightarrow \widetilde{H}}{G \| H \Rightarrow G \| \widetilde{H}} \qquad \frac{G \Rightarrow \widetilde{G}}{G[f] \Rightarrow \widetilde{G}[f]}$$

$$\frac{G \Rightarrow \widetilde{G},\ \circ \in \{\mathsf{rs}, \mathsf{sy}\}}{G \circ a \Rightarrow \widetilde{G} \circ a} \qquad \frac{G \Rightarrow \widetilde{G}}{[G * E * F] \Rightarrow [\widetilde{G} * E * F]} \qquad \frac{G \Rightarrow \widetilde{G}}{[E * G * F] \Rightarrow [E * \widetilde{G} * F]} \qquad \frac{G \Rightarrow \widetilde{G}}{[E * F * G] \Rightarrow [E * F * \widetilde{G}]}$$

An *operative regular dynamic expression* G: no inaction rule can be applied to it.

$OpRegDynExpr$ is the set of *all operative regular dynamic expressions* of $dtsPBC$.

We shall consider regular expressions only and omit the word "regular".

Definition 4 $\approx\ = (\Rightarrow \cup \Leftarrow)^*$ *is the structural equivalence of dynamic expressions in* $dtsPBC$*.* G *and* G' *are* structurally equivalent, $G \approx G'$*, if they can be reached each from other by applying inaction rules in forward or backward direction.*

Action and empty loop rules

Action rules: execution of non-empty multisets of activities at a time step.

Empty loop rule: execution of the empty multiset of activities at a time step.

Let $(\alpha,\rho),(\beta,\chi)\in\mathcal{SL},\ E,F\in RegStatExpr,\ G,H\in OpRegDynExpr,$
$\widetilde{G},\widetilde{H}\in RegDynExpr,\ a\in Act$ and $\Gamma,\Delta\in I\!\!N_f^{\mathcal{SL}}\setminus\{\emptyset\},\ \Gamma'\in I\!\!N_f^{\mathcal{SL}}$.

Action and empty loop rules

El $G\stackrel{\emptyset}{\rightarrow}G$

B $\overline{(\alpha,\rho)}\stackrel{\{(\alpha,\rho)\}}{\longrightarrow}\underline{(\alpha,\rho)}$

SC1 $\dfrac{G\stackrel{\Gamma}{\rightarrow}\widetilde{G},\ \circ\in\{;,[]\}}{G\circ E\stackrel{\Gamma}{\rightarrow}\widetilde{G}\circ E}$

SC2 $\dfrac{G\stackrel{\Gamma}{\rightarrow}\widetilde{G},\ \circ\in\{;,[]\}}{E\circ G\stackrel{\Gamma}{\rightarrow}E\circ\widetilde{G}}$

P1 $\dfrac{G\stackrel{\Gamma}{\rightarrow}\widetilde{G}}{G\|H\stackrel{\Gamma}{\rightarrow}\widetilde{G}\|H}$

P2 $\dfrac{H\stackrel{\Gamma}{\rightarrow}\widetilde{H}}{G\|H\stackrel{\Gamma}{\rightarrow}G\|\widetilde{H}}$

P3 $\dfrac{G\stackrel{\Gamma}{\rightarrow}\widetilde{G},\ H\stackrel{\Delta}{\rightarrow}\widetilde{H}}{G\|H\stackrel{\Gamma+\Delta}{\longrightarrow}\widetilde{G}\|\widetilde{H}}$

L $\dfrac{G\stackrel{\Gamma}{\rightarrow}\widetilde{G}}{G[f]\stackrel{f(\Gamma)}{\longrightarrow}\widetilde{G}[f]}$

Rs $\dfrac{G\stackrel{\Gamma}{\rightarrow}\widetilde{G},\ a,\hat{a}\not\in\mathcal{A}(\Gamma)}{G\ \text{rs}\ a\stackrel{\Gamma}{\rightarrow}\widetilde{G}\ \text{rs}\ a}$

I1 $\dfrac{G\stackrel{\Gamma}{\rightarrow}\widetilde{G}}{[G*E*F]\stackrel{\Gamma}{\rightarrow}[\widetilde{G}*E*F]}$

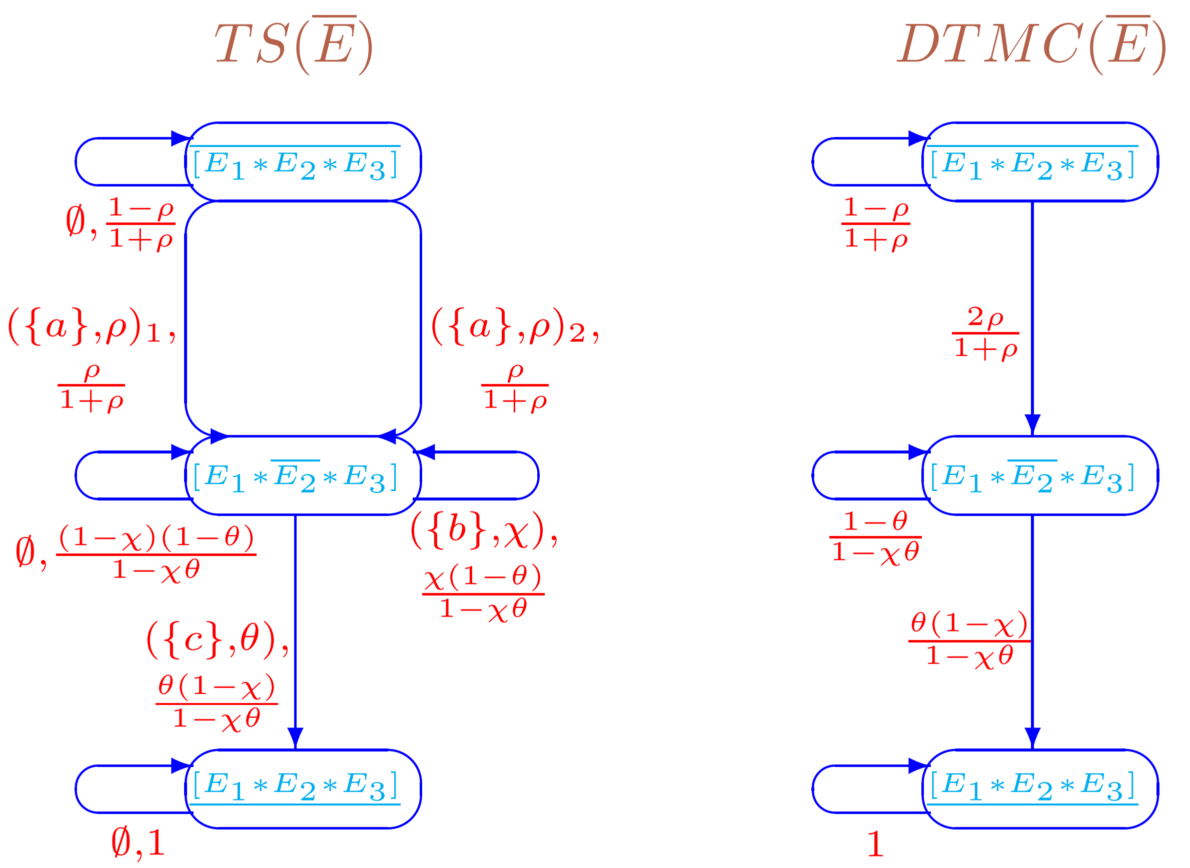
I2 $\dfrac{G\stackrel{\Gamma}{\rightarrow}\widetilde{G}}{[E*G*F]\stackrel{\Gamma}{\rightarrow}[E*\widetilde{G}*F]}$

I3 $\dfrac{G\stackrel{\Gamma}{\rightarrow}\widetilde{G}}{[E*F*G]\stackrel{\Gamma}{\rightarrow}[E*F*\widetilde{G}]}$

Sy1 $\dfrac{G\stackrel{\Gamma}{\rightarrow}\widetilde{G}}{G\ \text{sy}\ a\stackrel{\Gamma}{\rightarrow}\widetilde{G}\ \text{sy}\ a}$

Sy2 $\dfrac{G\ \text{sy}\ a\xrightarrow{\Gamma'+\{(\alpha,\rho)\}+\{(\beta,\chi)\}}\widetilde{G}\ \text{sy}\ a,\ a\in\alpha,\ \hat{a}\in\beta}{G\ \text{sy}\ a\xrightarrow{\Gamma'+\{(\alpha\oplus_a\beta,\rho\cdot\chi)\}}\widetilde{G}\ \text{sy}\ a}$

Transition systems



EXPRIT:The transition system and the underlying DTMC of $\overline{E}$ for $E = [((\{a\},\rho)_1[](\{a\},\rho)_2) * (\{b\},\chi) * (\{c\},\theta)]$

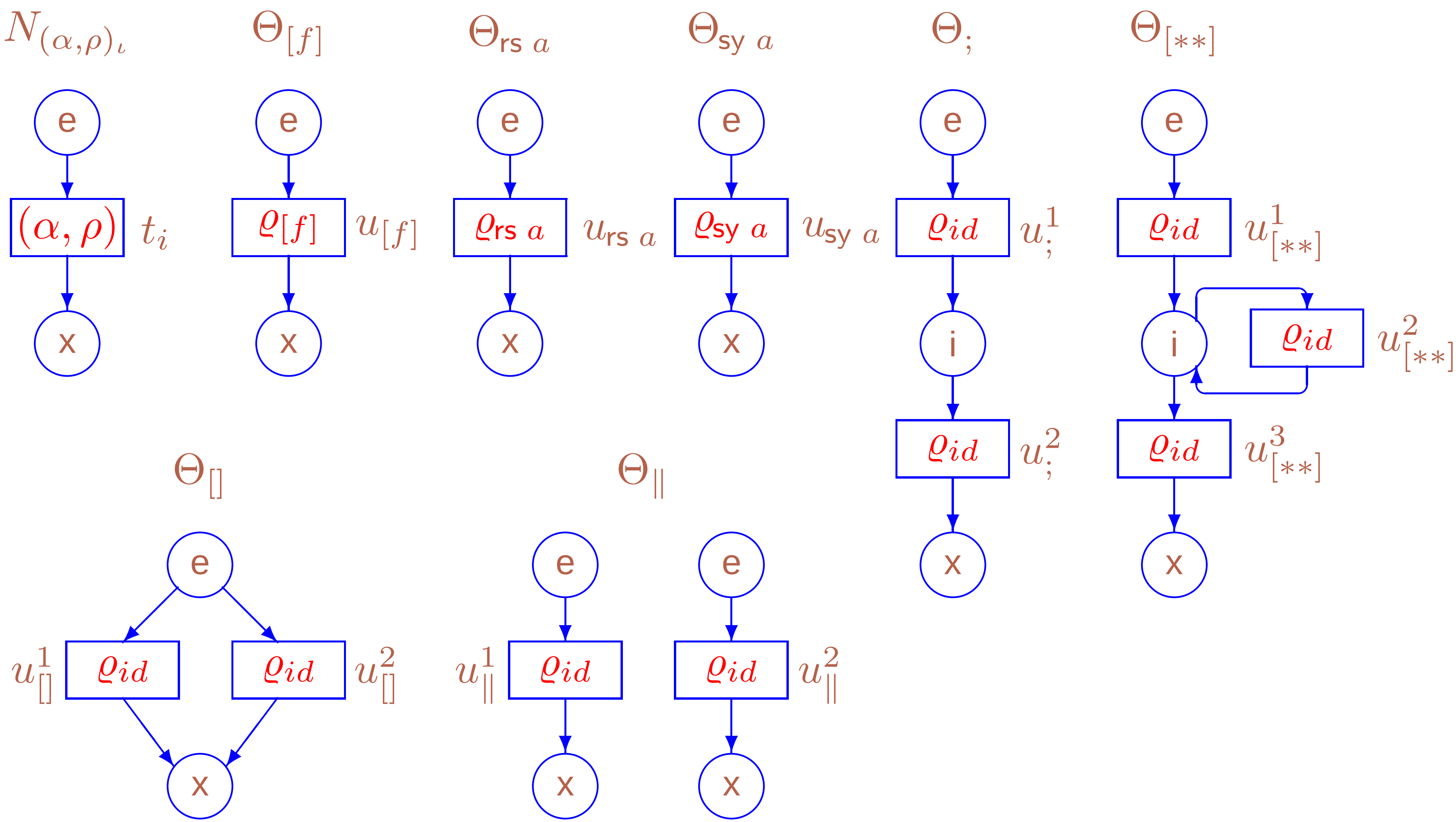
Let $E_1 = (\{a\},\rho)[](\{a\},\rho),\ E_2 = (\{b\},\chi),\ E_3 = (\{c\},\theta)$ and $E = [E_1 * E_2 * E_3]$.

The identical activities of the composite static expression are enumerated as:

$E = [((\{a\},\rho)_1[](\{a\},\rho)_2) * (\{b\},\chi) * (\{c\},\theta)]$. The derivation set $DR(\overline{E})$ of $\overline{E}$ consists of

$s_1 = [\overline{[E_1 * E_2 * E_3]}]_\approx,\ s_2 = [[E_1 * \overline{E_2} * E_3]]_\approx,\ s_3 = [\underline{[E_1 * E_2 * E_3]}]_\approx$.

Denotational semantics



The plain and operator dts-boxes

Definition **5** *Let* $(\alpha, \rho) \in \mathcal{SL},\ a \in Act$ *and* $E, F, K \in RegStatExpr$. *The* denotational semantics *of* $dtsPBC$ *is a mapping* Box_{dts} *from* $RegStatExpr$ *into plain dts-boxes:*

1. $Box_{dts}((\alpha, \rho)_\iota) = N_{(\alpha,\rho)_\iota}$;
2. $Box_{dts}(E \circ F) = \Theta_\circ(Box_{dts}(E), Box_{dts}(F)),\ \circ \in \{;, [], \|\}$;
3. $Box_{dts}(E[f]) = \Theta_{[f]}(Box_{dts}(E))$;
4. $Box_{dts}(E \circ a) = \Theta_{\circ a}(Box_{dts}(E)),\ \circ \in \{\mathrm{rs}, \mathrm{sy}\}$;
5. $Box_{dts}([E * F * K]) = \Theta_{[**]}(Box_{dts}(E), Box_{dts}(F), Box_{dts}(K))$.

For $E \in RegStatExpr$, let $Box_{dts}(\overline{E}) = \overline{Box_{dts}(E)}$ and $Box_{dts}(\underline{E}) = \underline{Box_{dts}(E)}$.

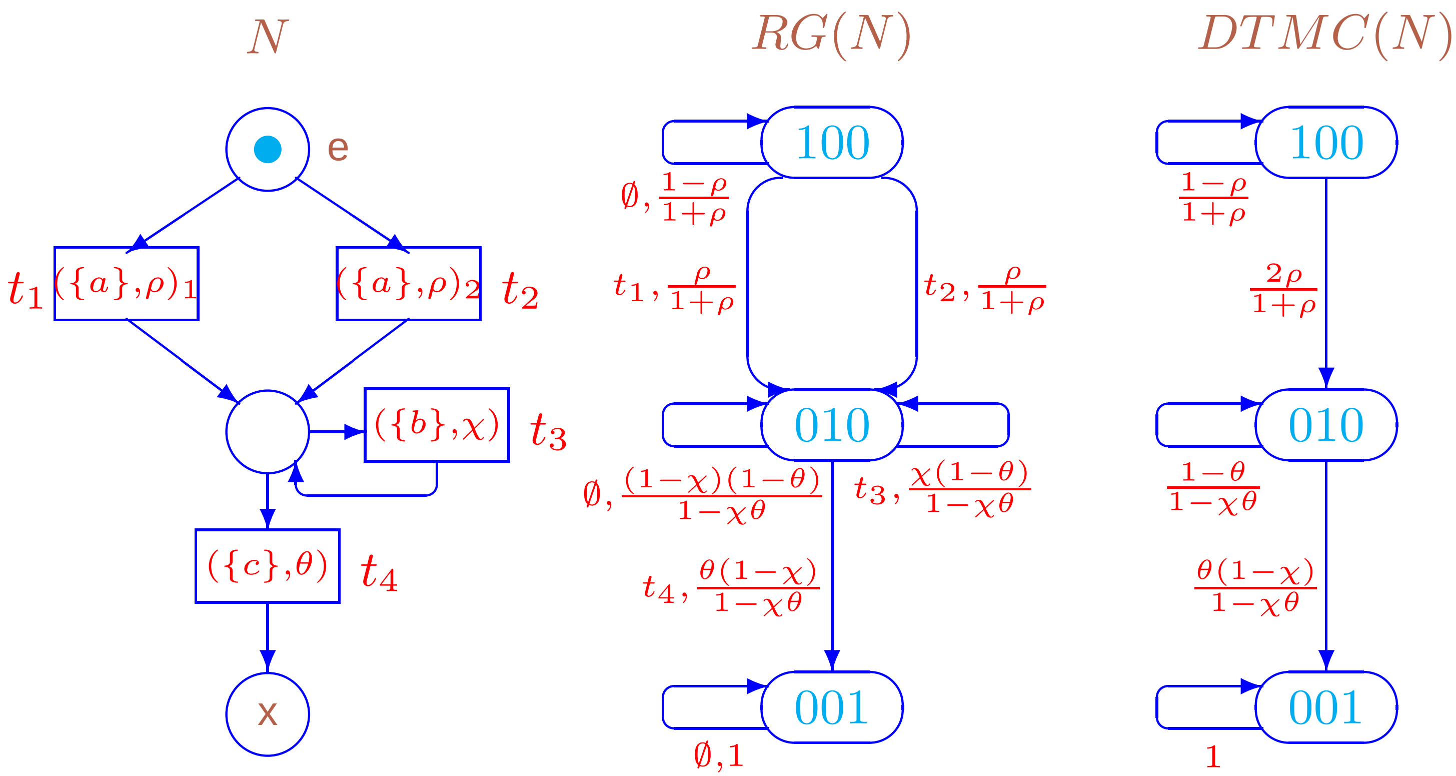
We denote isomorphism of transition systems by $\simeq$,

and the same symbol denotes isomorphism of reachability graphs and DTMCs

as well as isomorphism between transition systems and reachability graphs.

Theorem **1** *For any static expression* E *we have* $TS(\overline{E}) \simeq RG(Box_{dts}(\overline{E}))$.

Proposition **1** *For any static expression* E *we have* $DTMC(\overline{E}) \simeq DTMC(Box_{dts}(\overline{E}))$.



BOXIT:The marked dts-box $N = Box_{dts}(\overline{E})$ for $E = [((\{a\},\rho)_1[](\{a\},\rho)_2) * (\{b\},\chi) * (\{c\},\theta)]$, its reachability graph and the underlying DTMC

Performance evaluation

The elements $\mathcal{P}_{ij}$ $(1 \leq i, j \leq n = |DR(G)|)$ of *(one-step) transition probability matrix (TPM)* $\mathbf{P}$ for $DTMC(G)$:

$$\mathcal{P}_{ij} = \begin{cases} PM(s_i, s_j), & s_i \to s_j; \\ 0, & \text{otherwise}. \end{cases}$$

The *transient (k-step, $k \in \mathbb{N}$) probability mass function (PMF)* $\psi[k] = (\psi_1[k], \ldots, \psi_n[k])$ for $DTMC(G)$ is the solution of $\psi[k] = \psi[0]\mathbf{P}^k$,

where $\psi[0] = (\psi_1[0], \ldots, \psi_n[0])$ is the *initial PMF*: $\psi_i[0] = \begin{cases} 1, & s_i = [G]_\approx; \\ 0, & \text{otherwise}. \end{cases}$

We have $\psi[k+1] = \psi[k]\mathbf{P},\ k \in \mathbb{N}$.

The *steady-state PMF* $\psi = (\psi_1, \ldots, \psi_n)$ for $DTMC(G)$ is the solution of $\begin{cases} \psi(\mathbf{P} - \mathbf{E}) = \mathbf{0} \\ \psi\mathbf{1}^T = 1 \end{cases}$,

where $\mathbf{0}$ is a vector with n values 0, $\mathbf{1}$ is that with n values 1.

When $DTMC(G)$ has the single steady state, $\psi = \lim_{k\to\infty} \psi[k]$.

For $s \in DR(G)$ with $s = s_i$ $(1 \le i \le n)$ we define $\psi[k](s) = \psi_i[k]$ $(k \in I\!N)$ and $\psi(s) = \psi_i$.

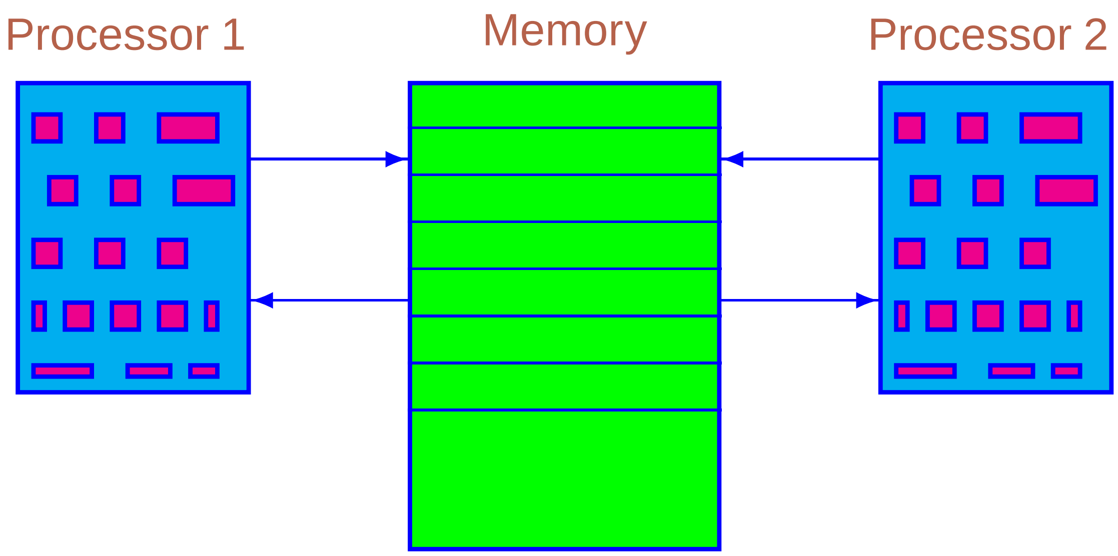
Let G be a dynamic expression and $s, \tilde{s} \in DR(G),\ S, \widetilde{S} \subseteq DR(G)$.

The following performance indices (measures) are based on the steady-state PMF.

- The *average recurrence (return) time in the state* s (the number of discrete time units or steps required for this) is $\frac{1}{\psi(s)}$.
- The *fraction of residence time in the state* s is $\psi(s)$.
- The *fraction of residence time in the set of states* $S \subseteq DR(G)$ or the *probability of the event determined by a condition that is true for all states from* S is $\sum_{s \in S} \psi(s)$.
- The *relative fraction of residence time in the set of states* S *w.r.t. that in* $\widetilde{S}$ is $\frac{\sum_{s \in S} \psi(s)}{\sum_{\tilde{s} \in \widetilde{S}} \psi(\tilde{s})}$.
- The *steady-state probability to perform a step with an activity* (α, ρ) is $\sum_{s \in DR(G)} \psi(s) \sum_{\{\Gamma \mid (\alpha,\rho) \in \Gamma\}} PT(\Gamma, s)$.
- The *probability of the event determined by a reward function* r *on the states* is $\sum_{s \in DR(G)} \psi(s) r(s)$.

Shared memory system

A model of two processors accessing a common shared memory [MBCDF95]



The diagram of the shared memory system

After activation of the system, two processors are active, and the common memory is available. Each processor can request an access to the memory.

When a processor starts an acquisition of the memory, another processor waits until the former one ends its operations, and the system returns to the state with both active processors and the available memory.

a corresponds to the system activation.

r_i $(1 \leq i \leq 2)$ represent the common memory request of processor i.

b_i and e_i correspond to the beginning and the end of the common memory access of processor i.

The other actions are used for communication purpose only.

The static expression of the first processor is

$E_1 = [(\{x_1\}, \frac{1}{2}) * ((\{r_1\}, \frac{1}{2}); (\{b_1, y_1\}, \frac{1}{2}); (\{e_1, z_1\}, \frac{1}{2})) * \mathsf{Stop}]$.

The static expression of the second processor is

$E_2 = [(\{x_2\}, \frac{1}{2}) * ((\{r_2\}, \frac{1}{2}); (\{b_2, y_2\}, \frac{1}{2}); (\{e_2, z_2\}, \frac{1}{2})) * \mathsf{Stop}]$.

The static expression of the shared memory is

$E_3 = [(\{a, \widehat{x_1}, \widehat{x_2}\}, \frac{1}{2}) * (((\{\widehat{y_1}\}, \frac{1}{2}); (\{\widehat{z_1}\}, \frac{1}{2}))[]((\{\widehat{y_2}\}, \frac{1}{2}); (\{\widehat{z_2}\}, \frac{1}{2}))) * \mathsf{Stop}]$.

The static expression of the shared memory system with two processors is

$E = (E_1 \| E_2 \| E_3)$ sy x_1 sy x_2 sy y_1 sy y_2 sy z_1 sy z_2 rs x_1 rs x_2 rs y_1 rs y_2 rs z_1 rs z_2.

Interpretation of the states

s_1: the initial state,

s_2: the system is activated and the memory is not requested,

s_3: the memory is requested by the first processor,

s_4: the memory is requested by the second processor,

s_5: the memory is allocated to the first processor,

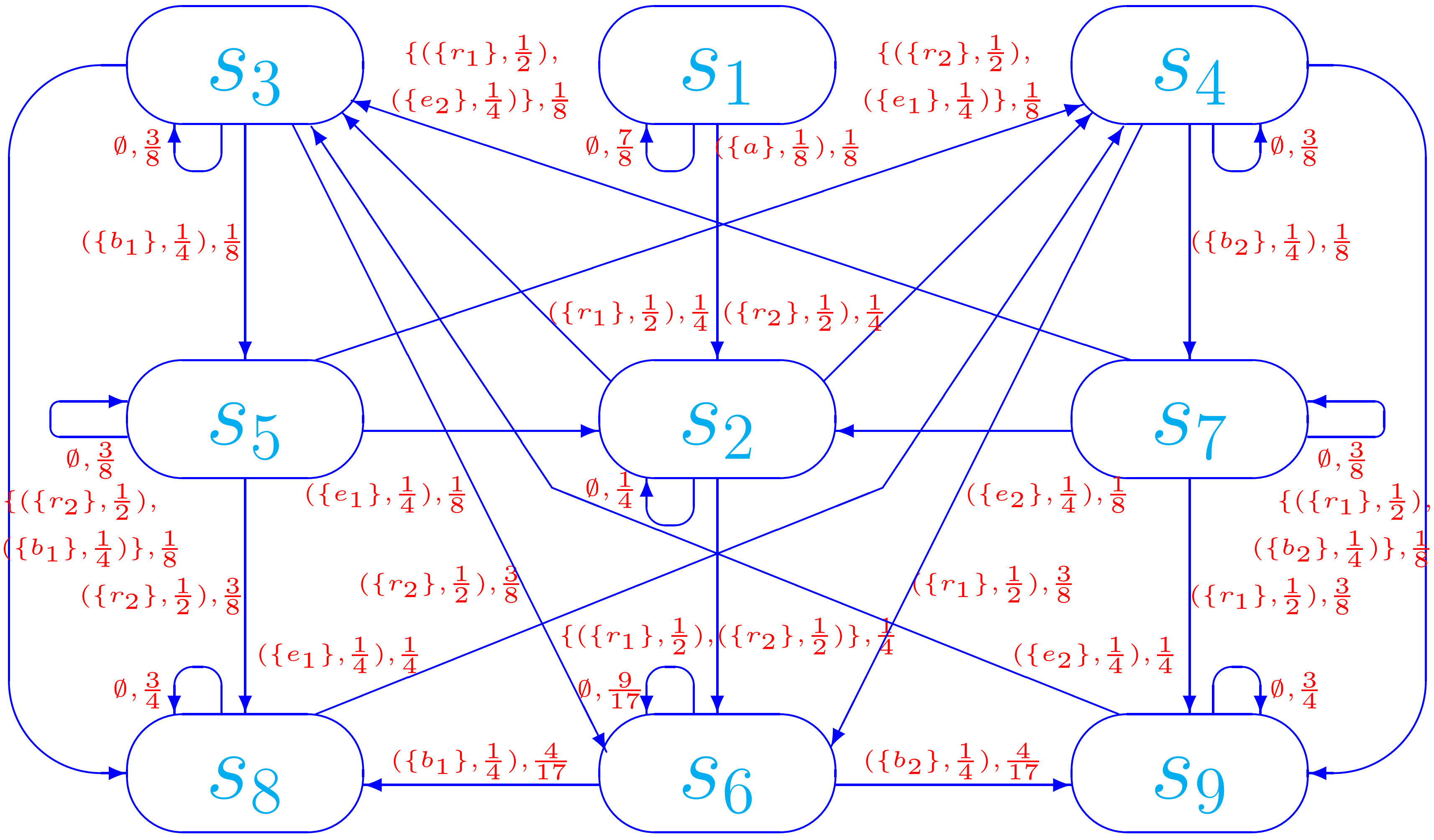
s_6: the memory is requested by two processors,

s_7: the memory is allocated to the second processor,

s_8: the memory is allocated to the first processor and the memory is requested by the second processor,

s_9: the memory is allocated to the second processor and the memory is requested by the first processor.

$TS(\overline{E})$



The transition system of the shared memory system

The TPM for $DTMC(\overline{E})$ is

$$\mathbf{P} = \begin{bmatrix} \frac{7}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & \frac{3}{8} & 0 & \frac{1}{8} & \frac{3}{8} & 0 & \frac{1}{8} & 0 \\ 0 & 0 & 0 & \frac{3}{3} & 0 & \frac{3}{8} & \frac{1}{8} & 0 & \frac{1}{8} \\ 0 & \frac{1}{8} & 0 & \frac{1}{8} & \frac{3}{8} & 0 & 0 & \frac{3}{8} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{9}{17} & 0 & \frac{4}{17} & \frac{4}{17} \\ 0 & \frac{1}{8} & \frac{1}{8} & 0 & 0 & 0 & \frac{3}{8} & 0 & \frac{3}{8} \\ 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{3}{4} & 0 \\ 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & \frac{3}{4} \end{bmatrix}.$$

The steady-state PMF for $DTMC(\overline{E})$ is

$$\psi = \left(0, \frac{16}{2103}, \frac{80}{701}, \frac{80}{701}, \frac{16}{701}, \frac{391}{2103}, \frac{16}{701}, \frac{560}{2103}, \frac{560}{2103}\right).$$

The average sojourn time vector of $\overline{E}$ is

$$SJ = \left(8, \frac{4}{3}, \frac{8}{5}, \frac{8}{5}, \frac{8}{5}, \frac{17}{8}, \frac{8}{5}, 4, 4\right).$$

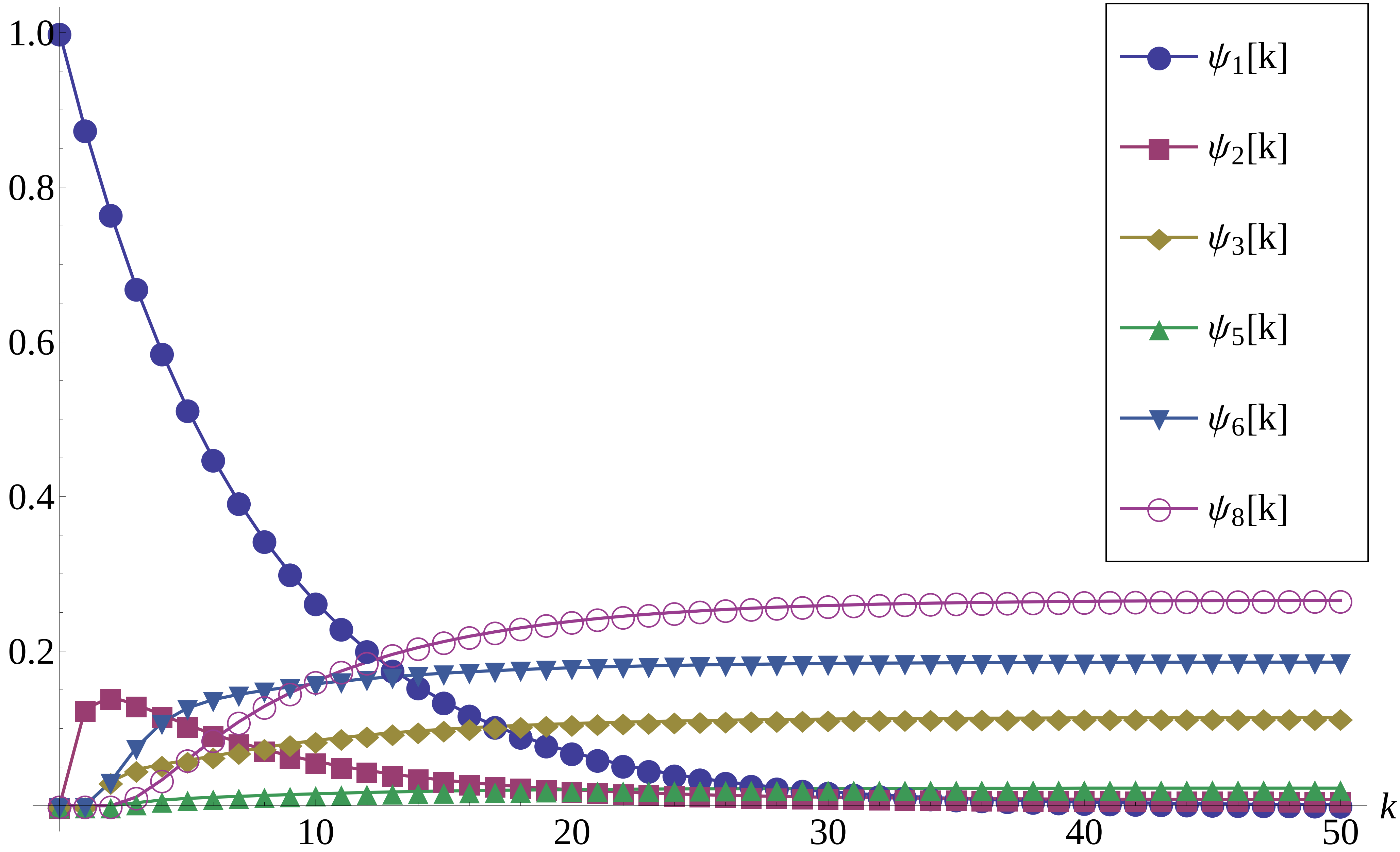
The sojourn time variance vector of $\overline{E}$ is

$$VAR = \left(56, \frac{4}{9}, \frac{24}{25}, \frac{24}{25}, \frac{24}{25}, \frac{153}{64}, \frac{24}{25}, 12, 12\right).$$

Transient and steady-state probabilities of the shared memory system

k	0	5	10	15	20	25	30	35	40	45	50	∞
$\psi_1[k]$	1	0.5129	0.2631	0.1349	0.0692	0.0355	0.0182	0.0093	0.0048	0.0025	0.0013	0
$\psi_2[k]$	0	0.1045	0.0573	0.0331	0.0207	0.0143	0.0110	0.0094	0.0085	0.0081	0.0078	0.0076
$\psi_3[k]$	0	0.0587	0.0845	0.0989	0.1063	0.1101	0.1121	0.1131	0.1136	0.1138	0.1140	0.1141
$\psi_5[k]$	0	0.0094	0.0154	0.0190	0.0209	0.0218	0.0223	0.0226	0.0227	0.0228	0.0228	0.0228
$\psi_6[k]$	0	0.1265	0.1577	0.1714	0.1785	0.1821	0.1840	0.1849	0.1854	0.1857	0.1858	0.1859
$\psi_8[k]$	0	0.0599	0.1611	0.2123	0.2386	0.2521	0.2590	0.2626	0.2644	0.2653	0.2658	0.2663

We depict the probabilities for the states $s_1, s_2, s_3, s_5, s_6, s_8$ only, since the corresponding values coincide for s_3, s_4 as well as for s_5, s_7 and for s_8, s_9.



Transient probabilities alteration diagram of the shared memory system

Performance indices

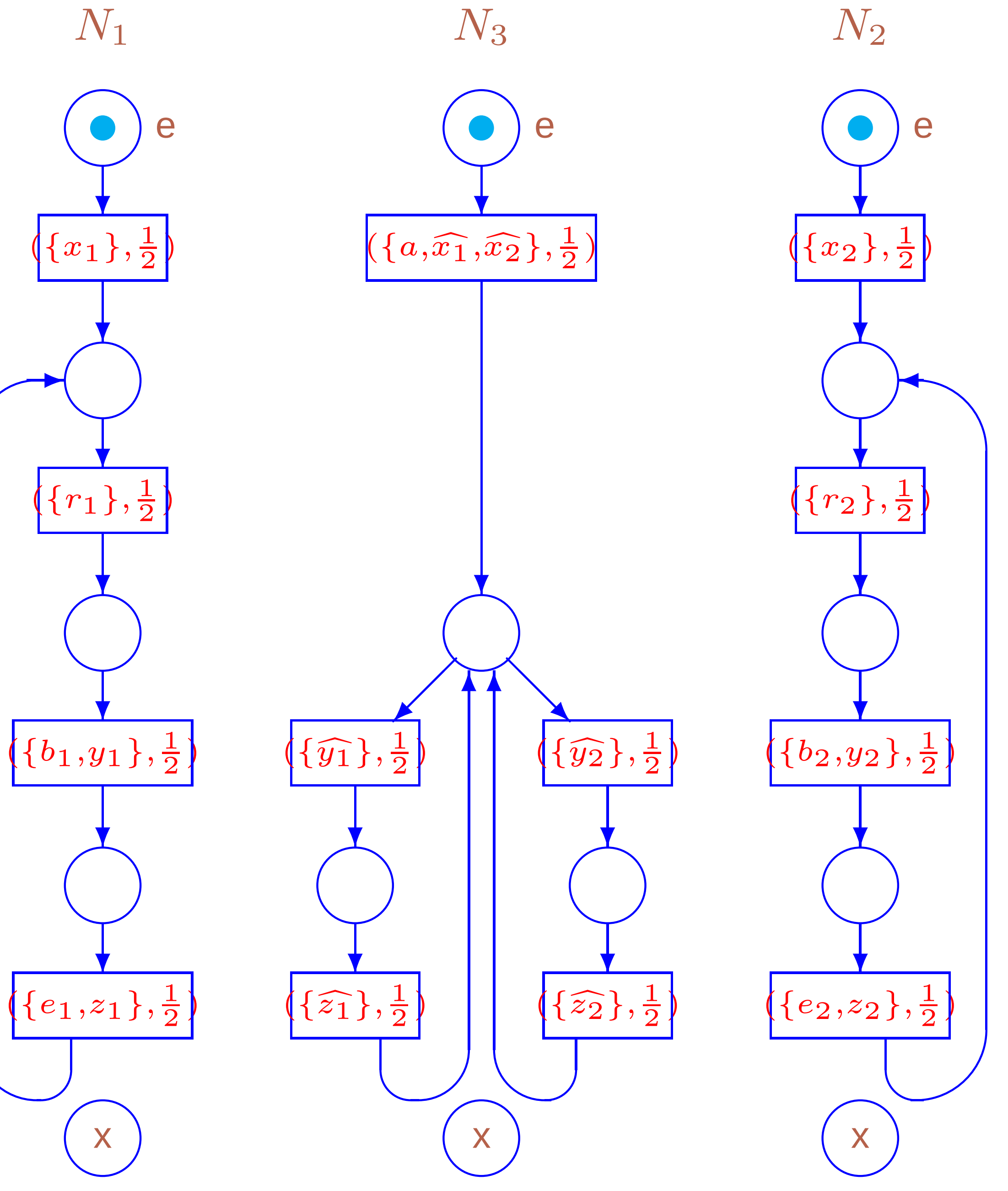
- The average recurrence time in the state s_2, the *average system run-through*, is $\frac{1}{\psi_2} = \frac{2103}{16} = 131\frac{7}{16}$.
- The common memory is available in the states s_2, s_3, s_4, s_6 only.

  The steady-state probability that the memory is available is $\psi_2 + \psi_3 + \psi_4 + \psi_6 = \frac{887}{2103}$.

  The steady-state probability that the memory is used, the *shared memory utilization*, is $1 - \frac{887}{2103} = \frac{1216}{2103}$.
- The common memory request of the first processor $(\{r_1\}, \frac{1}{2})$ is only possible from the states s_2, s_4, s_7.

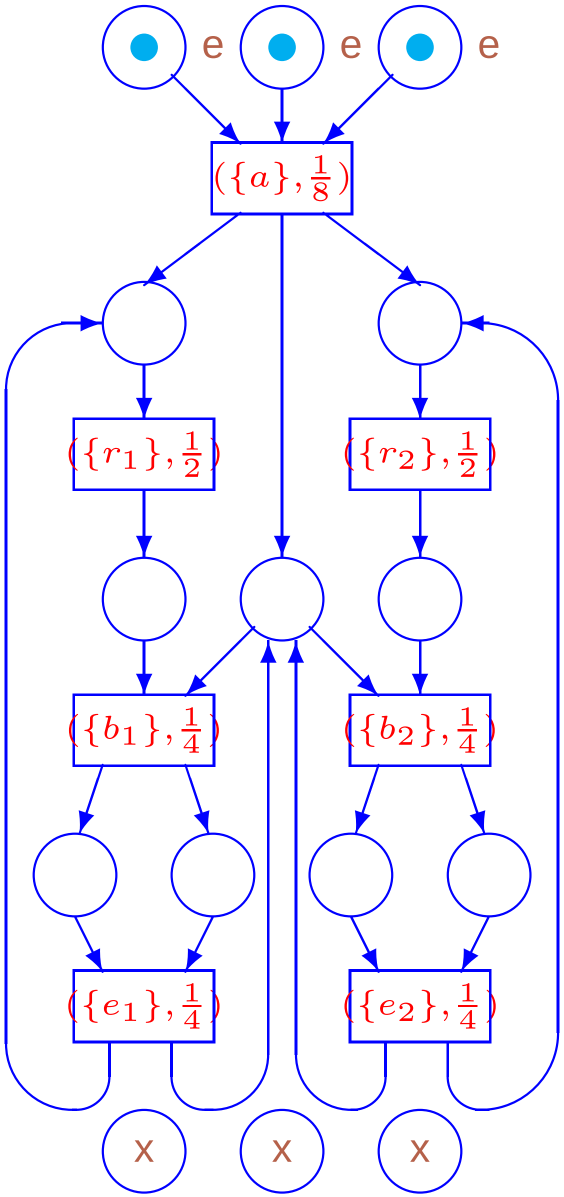
  The request probability in each of the states is a sum of execution probabilities for all multisets of activities containing $(\{r_1\}, \frac{1}{2})$.

  The *steady-state probability of the shared memory request from the first processor* is
  $\psi_2 \sum_{\{\Gamma|(\{r_1\},\frac{1}{2})\in\Gamma\}} PT(\Gamma, s_2) + \psi_4 \sum_{\{\Gamma|(\{r_1\},\frac{1}{2})\in\Gamma\}} PT(\Gamma, s_4) +$
  $\psi_7 \sum_{\{\Gamma|(\{r_1\},\frac{1}{2})\in\Gamma\}} PT(\Gamma, s_7) =$
  $\frac{16}{2103}\left(\frac{1}{4} + \frac{1}{4}\right) + \frac{80}{701}\left(\frac{3}{8} + \frac{1}{8}\right) + \frac{16}{701}\left(\frac{3}{8} + \frac{1}{8}\right) = \frac{152}{2103}$.



The marked dts-boxes of two processors and shared memory

N



The marked dts-box of the shared memory system

Overview and open questions

The results obtained

- A discrete time stochastic extension $dtsPBC$ of finite PBC enriched with iteration.
- A case study of performance analysis: the shared memory system.

Further research

- Defining stochastic equivalences to identify stochastic processes with similar behaviour.
- Introducing the deterministically timed multiactions with fixed time delays (including the zero delay).
- Extending the syntax with recursion operator.

References

[BDH92] BEST E., DEVILLERS R., HALL J.G. *The box calculus: a new causal algebra with multi-label communication. LNCS* **609**, p. 21–69, 1992.

[FM03] DE FRUTOS D.E., MARROQUÍN O.A. *Ambient Petri nets. Electronic Notes in Theoretical Computer Science* **85(1)**, 27 p., 2003.

[Kou00] KOUTNY M. *A compositional model of time Petri nets. LNCS* **1825**, p. 303–322, 2000.

[MBCDF95] MARSAN M.A., BALBO G., CONTE G., DONATELLI S., FRANCESCHINIS G. *Modelling with generalized stochastic Petri nets. Wiley Series in Parallel Computing*, John Wiley and Sons, 1995.

[MF00] MARROQUÍN O.A., DE FRUTOS D.E. *TPBC: timed Petri box calculus. Technical Report*, Departamento de Sistemas Infofmáticos y Programación, UCM, Madrid, Spain, 2000 (in Spanish).

[MVCC03] MACIÀ H.S., VALERO V.R., CAZORLA D.L., CUARTERO F.G. *Introducing the iteration in sPBC. Technical Report* **DIAB-03-01-37**, 20 p., Department of Computer Science, University of Castilla-La Mancha, Albacete, Spain, September 2003.

[MVCR08] MACIÀ H.S., VALERO V.R., CUARTERO F.G., RUIZ M.C.D. *sPBC: a Markovian extension of Petri box calculus with immediate multiactions. Fundamenta Informaticae* **87(3–4)**, p. 367–406, IOS Press, Amsterdam, The Netherlands, 2008.

[MVF01] MACIÀ H.S., VALERO V.R., DE FRUTOS D.E. *sPBC: a Markovian extension of finite Petri box calculus. Proceedings of 9^{th} IEEE International Workshop on Petri Nets and Performance Models - 01 (PNPM'01)*, p. 207–216, Aachen, Germany, IEEE Computer Society Press, September 2001.

[Nia05] NIAOURIS A. *An algebra of Petri nets with arc-based time restrictions. LNCS* **3407**, p. 447–462, 2005.

[Tar05] TARASYUK I.V. *Discrete time stochastic Petri box calculus. Berichte aus dem Department für Informatik* **3/05**, 25 p., Carl von Ossietzky Universität Oldenburg, Germany, November 2005, http://db.iis.nsk.su/persons/itar/dtspbcib_cov.pdf.

[Tar06] TARASYUK I.V. *Iteration in discrete time stochastic Petri box calculus. Bulletin of the Novosibirsk Computing Center, Series Computer Science, IIS Special Issue* **24**, p. 129–148, NCC Publisher, Novosibirsk, 2006, http://db.iis.nsk.su/persons/itar/dtsitncc.pdf.

[TMV10] TARASYUK I.V., MACIÀ H.S., VALERO V.R. *Discrete time stochastic Petri box calculus with immediate multiactions. Technical Report* **DIAB-10-03-1**, 25 p., Department of Computer Systems, High School of Computer Science Engineering, University of Castilla-La Mancha, Albacete, Spain, March 2010, http://www.dsi.uclm.es/descargas/thecnicalreports/DIAB-10-03-1/dtsipbc.pdf.

The slides can be downloaded from Internet:

`http://itar.iis.nsk.su/files/itar/pages/oldb11sld.pdf`

Thank you for your attention!