Performance evaluation in dtsPBC *

Igor V. Tarasyuk[1]

A.P. Ershov Institute of Informatics Systems SB RAS, Novosibirsk, Russia
`itar@iis.nsk.su`

Abstract. Algebra dtsPBC is a discrete time stochastic extension of finite Petri box calculus (PBC) enriched with iteration. In this paper, within dtsPBC, a method of modeling and performance evaluation based on stationary behaviour analysis for concurrent computing systems is outlined applied to the shared memory system.
Keywords: stochastic process algebra, Petri box calculus, iteration, discrete time, stationary behaviour, performance evaluation.

1 Introduction

Algebraic process calculi is a well-known formal model for specification of computing systems and analysis of their behaviour. In such process algebras (PAs), systems and processes are specified by formulas, and verification of their properties is accomplished at a syntactic level via equivalences, axioms and inference rules. In the last decades, stochastic extensions of PAs were proposed. Stochastic process algebras (SPAs) do not just specify actions which can happen as usual process algebras (qualitative features), but they associate some quantitative parameters with actions (quantitative characteristics).

Petri box calculus (PBC) [1] is a flexible and expressive process algebra developed as a tool for specification of Petri nets structure and their interrelations. Its goal was also to propose a compositional semantics for high level constructs of concurrent programming languages in terms of elementary Petri nets. PBC has a step operational semantics in terms of labeled transition systems. Its denotational semantics was proposed in terms of a subclass of Petri nets (PNs) equipped with interface and considered up to isomorphism called Petri boxes.

A stochastic extension of PBC called stochastic Petri box calculus (sPBC) was proposed in [5]. Only a finite part of PBC was used for the stochastic enrichment, i.e., sPBC has neither refinement nor recursion nor iteration operations. The calculus has an interleaving operational semantics in terms of labeled transition systems. Its denotational semantics was defined in terms of a subclass of labeled continuous time stochastic PNs (LCTSPNs) called s-boxes. In [4], the iteration was added to sPBC.

In [7], a discrete time stochastic extension dtsPBC of finite PBC was presented. A step operational semantics of dtsPBC was constructed via labeled

* This work was supported in part by Deutsche Forschungsgemeinschaft (DFG), 436 RUS 113/1002/01, and Russian Foundation for Basic Research (RFBR), 09-01-91334

probabilistic transition systems. Its denotational semantics was defined based on a subclass of labeled discrete time stochastic PNs (LDTSPNs) called dts-boxes. A variety of probabilistic equivalences were proposed and their interrelations were studied. In [6], the iteration operator was added to dtsPBC.

In this paper, we use dtsPBC with iteration as a basic model. First, we present syntax of the calculus. Second, we describe its operational semantics in terms of labeled transition systems and denotational semantics based on a subclass of LDTSPNs. Stationary behaviour of infinite stochastic processes within dtsPBC is described. For a system with two processors and common shared memory the performance indices are calculated based on steady state probabilities.

The paper is organized as follows. The syntax of dtsPBC is presented in Section 2. Section 3 describes its operational semantics and Section 4 presents its denotational semantics. In Section 5, a performance evaluation of stochastic processes is presented applied to the shared memory system. The concluding Section 6 summarizes the results obtained and outlines research perspectives.

2 Syntax

In this section, we propose the syntax of discrete time stochastic PBC (dtsPBC).

We denote the *set of all finite multisets* over X by $\mathbb{N}_f^X$. Let $Act = \{a, b, \ldots\}$ be the set of *elementary actions*. Then $\widehat{Act} = \{\hat{a}, \hat{b}, \ldots\}$ is the set of *conjugated actions (conjugates)* such that $a \neq \hat{a}$ and $\hat{\hat{a}} = a$. Let $\mathcal{A} = Act \cup \widehat{Act}$ be the set of *all actions*, and $\mathcal{L} = \mathbb{N}_f^{\mathcal{A}}$ be the set of *all multiactions*. Note that $\emptyset \in \mathcal{L}$, this corresponds to the execution of a multiaction that contains no visible action names. The *alphabet* of $\alpha \in \mathcal{L}$ is defined as $\mathcal{A}(\alpha) = \{x \in \mathcal{A} \mid \alpha(x) > 0\}$.

An *activity (stochastic multiaction)* is a pair (α, ρ), where $\alpha \in \mathcal{L}$ and $\rho \in (0; 1)$ is the probability of the multiaction α. The multiaction probabilities are used to calculate probabilities of state changes (steps) at discrete time moments. Let $\mathcal{SL}$ be the set of *all activities*. Let us note that the same multiaction $\alpha \in \mathcal{L}$ may have different probabilities in the same specification. The *alphabet* of $(\alpha, \rho) \in \mathcal{SL}$ is defined as $\mathcal{A}(\alpha, \rho) = \mathcal{A}(\alpha)$. For $(\alpha, \rho) \in \mathcal{SL}$, we define its *multiaction part* as $\mathcal{L}(\alpha, \rho) = \alpha$ and its *probability part* as $\Omega(\alpha, \rho) = \rho$.

Activities are combined into formulas by the following operations: *sequential execution* ;, *choice* [], *parallelism* $\|$, *relabeling* $[f]$ of actions, *restriction* rs over a single action, *synchronization* sy on an action and its conjugate and *iteration* $[**]$ with three arguments: initialization, body and termination. First, the initialization subprocess is executed, then the body one is performed zero or more times, and, finally, the termination one is executed.

Relabeling functions $f : \mathcal{A} \to \mathcal{A}$ are bijections preserving conjugates, i.e., $\forall x \in \mathcal{A}\ f(\hat{x}) = \widehat{f(x)}$. Let $\alpha, \beta \in \mathcal{L}$ be two multiactions such that for some action $a \in Act$ we have $a \in \alpha$ and $\hat{a} \in \beta$ or $\hat{a} \in \alpha$ and $a \in \beta$. Then synchronization of α and β by a is defined as $\alpha \oplus_a \beta = \gamma$, where

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

Static expressions specify the structure of a system. They correspond to unmarked SPNs.

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

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

StatExpr denotes the set of *all static expressions* of dtsPBC.

To make the grammar above unambiguous, one can add parentheses in the productions with binary operations: $(E;E)$, $(E[]E)$, $(E\|E)$. However, here and further we prefer the PBC approach and add them to resolve ambiguities only.

To avoid inconsistency of the iteration operator, we do not allow concurrency in the highest level of the second argument of iteration. This is not a severe restriction, since we can prefix parallel expressions by an activity with the empty multiaction and an appropriate probability.

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

$$\begin{aligned} D &::= \ (\alpha,\rho) \mid D;E \mid D[]D \mid D[f] \mid D \ \mathsf{rs}\ a \mid D \ \mathsf{sy}\ a \mid [D*D*E],\\ E &::= \ (\alpha,\rho) \mid E;E \mid E[]E \mid E\|E \mid E[f] \mid E \ \mathsf{rs}\ a \mid E \ \mathsf{sy}\ a \mid [E*D*E].\end{aligned}$$

RegStatExpr denotes the set of *all regular static expressions* of dtsPBC.

Dynamic expressions specify the states of a system. They correspond to marked SPNs. $\overline{E}$ denotes the *initial*, and $\underline{E}$ denotes the *final* state of the process specified by a static expression E.

Definition 3. *Let* $(\alpha,\rho)\in\mathcal{SL}$, $a\in Act$ *and* $E\in RegStatExpr$*. A* regular dynamic expression *of dtsPBC is defined as*

$$\begin{aligned} G ::= \ &\overline{E} \mid \underline{E} \mid G;E \mid E;G \mid G[]E \mid E[]G \mid G\|G \mid G[f] \mid G \ \mathsf{rs}\ a \mid G \ \mathsf{sy}\ a \mid\\ &[G*E*E] \mid [E*G*E] \mid [E*E*G].\end{aligned}$$

RegDynExpr denotes the set of *all regular dynamic expressions* of dtsPBC. We shall consider regular expressions only and omit the word "regular".

3 Operational semantics

In this section, we construct the step operational semantics in terms of labeled probabilistic transition systems.

3.1 Inaction rules

Inaction rules describe expression transformations due to execution of empty multisets of activities, this does not change states of the corresponding processes.

First, we define inaction rules for overlined and underlined static expressions. Let $E, F, K \in \mathit{RegStatExpr}$ and $a \in \mathit{Act}$.

$$\begin{array}{lll}
\overline{E;F} \overset{\emptyset}{\rightarrow} \overline{E};F & \underline{E};F \overset{\emptyset}{\rightarrow} E;\overline{F} & E;\underline{F} \overset{\emptyset}{\rightarrow} \underline{E;F} \\
\overline{E[\,]F} \overset{\emptyset}{\rightarrow} \overline{E}[\,]F & \overline{E[\,]F} \overset{\emptyset}{\rightarrow} E[\,]\overline{F} & \underline{E}[\,]F \overset{\emptyset}{\rightarrow} \underline{E[\,]F} \\
E[\,]\underline{F} \overset{\emptyset}{\rightarrow} \underline{E[\,]F} & \overline{E\|F} \overset{\emptyset}{\rightarrow} \overline{E}\|\overline{F} & \underline{E}\|\underline{F} \overset{\emptyset}{\rightarrow} \underline{E\|F} \\
\overline{E[f]} \overset{\emptyset}{\rightarrow} \overline{E}[f] & \underline{E}[f] \overset{\emptyset}{\rightarrow} \underline{E[f]} & \overline{E \ \mathsf{rs}\ a} \overset{\emptyset}{\rightarrow} \overline{E} \ \mathsf{rs}\ a \\
\underline{E} \ \mathsf{rs}\ a \overset{\emptyset}{\rightarrow} \underline{E \ \mathsf{rs}\ a} & \overline{E \ \mathsf{sy}\ a} \overset{\emptyset}{\rightarrow} \overline{E} \ \mathsf{sy}\ a & \underline{E} \ \mathsf{sy}\ a \overset{\emptyset}{\rightarrow} \underline{E \ \mathsf{sy}\ a}
\end{array}$$

$$\overline{[E*F*K]} \overset{\emptyset}{\rightarrow} [\overline{E}*F*K] \quad [\underline{E}*F*K] \overset{\emptyset}{\rightarrow} [E*\overline{F}*K] \quad [E*\underline{F}*K] \overset{\emptyset}{\rightarrow} [E*\overline{F}*K]$$

$$[E*\underline{F}*K] \overset{\emptyset}{\rightarrow} [E*F*\overline{K}] \quad [E*F*\underline{K}] \overset{\emptyset}{\rightarrow} \underline{[E*F*K]}$$

Second, we propose inaction rules for dynamic expressions.
Let $E, F \in \mathit{RegStatExpr},\ G, H, \widetilde{G}, \widetilde{H} \in \mathit{RegDynExpr}$ and $a \in \mathit{Act}$.

$$G \overset{\emptyset}{\rightarrow} G \quad \frac{G \overset{\emptyset}{\rightarrow} \widetilde{G},\ \circ \in \{;, [\,]\}}{G \circ E \overset{\emptyset}{\rightarrow} \widetilde{G} \circ E} \quad \frac{G \overset{\emptyset}{\rightarrow} \widetilde{G},\ \circ \in \{;, [\,]\}}{E \circ G \overset{\emptyset}{\rightarrow} E \circ \widetilde{G}} \quad \frac{G \overset{\emptyset}{\rightarrow} \widetilde{G}}{G\|H \overset{\emptyset}{\rightarrow} \widetilde{G}\|H} \quad \frac{H \overset{\emptyset}{\rightarrow} \widetilde{H}}{G\|H \overset{\emptyset}{\rightarrow} G\|\widetilde{H}}$$

$$\frac{G \overset{\emptyset}{\rightarrow} \widetilde{G}}{G[f] \overset{\emptyset}{\rightarrow} \widetilde{G}[f]} \quad \frac{G \overset{\emptyset}{\rightarrow} \widetilde{G},\ \circ \in \{\mathsf{rs}, \mathsf{sy}\}}{G \circ a \overset{\emptyset}{\rightarrow} \widetilde{G} \circ a} \quad \frac{G \overset{\emptyset}{\rightarrow} \widetilde{G}}{[G*E*F] \overset{\emptyset}{\rightarrow} [\widetilde{G}*E*F]} \quad \frac{G \overset{\emptyset}{\rightarrow} \widetilde{G}}{[E*G*F] \overset{\emptyset}{\rightarrow} [E*\widetilde{G}*F]} \quad \frac{G \overset{\emptyset}{\rightarrow} \widetilde{G}}{[E*F*G] \overset{\emptyset}{\rightarrow} [E*F*\widetilde{G}]}$$

The rule $G \overset{\emptyset}{\rightarrow} G$ is intentionally included in the set of rules above. It reflects a non-zero probability to stay in a state at the next time moment, an essential feature of discrete time stochastic processes. The rule has no prototype in PBC.

A regular dynamic expression G is *operative* if no inaction rule can be applied to it, with the exception of $G \overset{\emptyset}{\rightarrow} G$. Note that any dynamic expression can be always transformed into a (not necessarily unique) operative one using inaction rules. Let *OpRegDynExpr* denote the set of *all operative regular dynamic expressions* of dtsPBC.

Definition 4. *Let* $\simeq\, = (\overset{\emptyset}{\rightarrow} \cup \overset{\emptyset}{\leftarrow})^*$ *be isomorphism of dynamic expressions in dtsPBC. Thus, two dynamic expressions* G *and* G' *are* isomorphic*, denoted by* $G \simeq G'$*, if they can be reached from each other by applying inaction rules.*

3.2 Action rules

Action rules describe expression transformations due to execution of arbitrary multisets of activities, this can change states of the corresponding processes.

We propose action rules describing expression transformations due to the execution of multisets of activities. Let $(\alpha, \rho), (\beta, \chi) \in \mathcal{SL},\ E, F \in \mathit{RegStatExpr}$, $G, H \in \mathit{OpRegDynExpr},\ \widetilde{G}, \widetilde{H} \in \mathit{RegDynExpr}$ and $a \in \mathit{Act}$. Moreover, let $\varGamma, \varDelta \in \mathbb{N}_f^{\mathcal{SL}}$. The *alphabet* of $\varGamma \in \mathbb{N}_f^{\mathcal{SL}}$ is defined as $\mathcal{A}(\varGamma) = \cup_{(\alpha,\rho)\in\varGamma} \mathcal{A}(\alpha)$.

$$\begin{array}{llllll}
\mathbf{B} & \overline{(\alpha,\rho)} \stackrel{\{(\alpha,\rho)\}}{\longrightarrow} \underline{(\alpha,\rho)} & \mathbf{SC1} & \frac{G\stackrel{\mathit{\Gamma}}{\to}\widetilde{G},\ \circ\in\{;,[]\}}{G\circ E\stackrel{\mathit{\Gamma}}{\to}\widetilde{G}\circ E} & \mathbf{SC2} & \frac{G\stackrel{\mathit{\Gamma}}{\to}\widetilde{G},\ \circ\in\{;,[]\}}{E\circ G\stackrel{\mathit{\Gamma}}{\to}E\circ\widetilde{G}} \\
\mathbf{P1} & \frac{G\stackrel{\mathit{\Gamma}}{\to}\widetilde{G}}{G\|H\stackrel{\mathit{\Gamma}}{\to}\widetilde{G}\|H} & \mathbf{P2} & \frac{H\stackrel{\mathit{\Gamma}}{\to}\widetilde{H}}{G\|H\stackrel{\mathit{\Gamma}}{\to}G\|\widetilde{H}} & \mathbf{P3} & \frac{G\stackrel{\mathit{\Gamma}}{\to}\widetilde{G},\ H\stackrel{\mathit{\Delta}}{\to}\widetilde{H}}{G\|H\stackrel{\mathit{\Gamma}+\mathit{\Delta}}{\longrightarrow}\widetilde{G}\|\widetilde{H}} \\
\mathbf{L} & \frac{G\stackrel{\mathit{\Gamma}}{\to}\widetilde{G}}{G[f]\stackrel{f(\mathit{\Gamma})}{\longrightarrow}\widetilde{G}[f]} & \mathbf{Rs} & \frac{G\stackrel{\mathit{\Gamma}}{\to}\widetilde{G},\ a,\hat{a}\not\in\mathcal{A}(\mathit{\Gamma})}{G \ \mathsf{rs}\ a\stackrel{\mathit{\Gamma}}{\to}\widetilde{G}\ \mathsf{rs}\ a} & \mathbf{I1} & \frac{G\stackrel{\mathit{\Gamma}}{\to}\widetilde{G}}{[G{*}E{*}F]\stackrel{\mathit{\Gamma}}{\to}[\widetilde{G}{*}E{*}F]} \\
\mathbf{I2} & \frac{G\stackrel{\mathit{\Gamma}}{\to}\widetilde{G}}{[E{*}G{*}F]\stackrel{\mathit{\Gamma}}{\to}[E{*}\widetilde{G}{*}F]} & \mathbf{I3} & \frac{G\stackrel{\mathit{\Gamma}}{\to}\widetilde{G}}{[E{*}F{*}G]\stackrel{\mathit{\Gamma}}{\to}[E{*}F{*}\widetilde{G}]} & \mathbf{Sy1} & \frac{G\stackrel{\mathit{\Gamma}}{\to}\widetilde{G}}{G\ \mathsf{sy}\ a\stackrel{\mathit{\Gamma}}{\to}\widetilde{G}\ \mathsf{sy}\ a}
\end{array}$$

$$\mathbf{Sy2}\ \frac{G\ \mathsf{sy}\ a\stackrel{\mathit{\Gamma}+\{(\alpha,\rho)\}+\{(\beta,\chi)\}}{\longrightarrow}\widetilde{G}\ \mathsf{sy}\ a,\ a\in\mathcal{A}(\alpha),\ \hat{a}\in\mathcal{A}(\beta)}{G\ \mathsf{sy}\ a\stackrel{\mathit{\Gamma}+\{(\alpha\oplus_a\beta,\rho\cdot\chi)\}}{\longrightarrow}\widetilde{G}\ \mathsf{sy}\ a}$$

In the rule **Sy2** we multiply the probabilities of synchronized multiactions, it corresponds to the event intersection. The rule has no analogue in PBC.

3.3 Transition systems

Now we define labeled probabilistic transition systems associated with dynamic expressions and used to define their operational semantics. Note that expressions can contain identical activities. To avoid technical difficulties, we can always enumerate coinciding activities from left to right in the syntax of expressions.

Definition 5. *Let G be a dynamic expression. Then $[G]_\simeq = \{H \mid G \simeq H\}$ is the equivalence class of G w.r.t. isomorphism (the isomorphism class).*

The derivation set *of a dynamic expression G, denoted by $DR(G)$, is the minimal set such that $[G]_\simeq \in DR(G)$ or, if $[H]_\simeq \in DR(G)$ and $\exists \mathit{\Gamma}\ H \stackrel{\mathit{\Gamma}}{\to} \widetilde{H}$, then $[\widetilde{H}]_\simeq \in DR(G)$.*

Let G be a dynamic expression and $s \in DR(G)$. The set of *all multisets of activities executable in* s is defined as $Exec(s) = \{\mathit{\Gamma} \mid \exists H \in s\ \exists \widetilde{H}\ H \stackrel{\mathit{\Gamma}}{\to} \widetilde{H}\}$.

The probability that the activities from $\mathit{\Gamma} \in Exec(s)$ *try to happen* in s is

$$PF(\mathit{\Gamma}, s) = \prod_{(\alpha,\rho)\in\mathit{\Gamma}} \rho \cdot \prod_{\{\{(\beta,\chi)\}\in Exec(s) \mid (\beta,\chi)\not\in\mathit{\Gamma}\}} (1-\chi).$$

In the case $\mathit{\Gamma} = \emptyset$ we define

$$PF(\emptyset, s) = \begin{cases} \prod_{\{(\beta,\chi)\}\in Exec(s)} (1-\chi), & Exec(s) \neq \{\emptyset\}; \\ 1, & \text{otherwise.} \end{cases}$$

The probability that the activities from $\mathit{\Gamma} \in Exec(s)$ *happen* in s is

$$PT(\mathit{\Gamma}, s) = \frac{PF(\mathit{\Gamma}, s)}{\sum_{\mathit{\Delta}\in Exec(s)} PF(\mathit{\Delta}, s)}.$$

The probability that the execution of *any* activities changes s to $\tilde{s}$ is

$$PM(s, \tilde{s}) = \sum_{\{\mathit{\Gamma} \mid \exists H\in s\ \exists \widetilde{H}\in\tilde{s}\ H\stackrel{\mathit{\Gamma}}{\to}\widetilde{H}\}} PT(\mathit{\Gamma}, s).$$

Definition 6. *Let G be a dynamic expression. The* (labeled probabilistic) transition system *of G is a quadruple $TS(G) = (S_G, L_G, \mathcal{T}_G, s_G)$, where*

- *the set of* states *is $S_G = DR(G)$;*
- *the set of* labels *is $L_G \subseteq I\!N_f^{\mathcal{SL}} \times (0;1]$;*
- *the set of* transitions *is $\mathcal{T}_G = \{(s, (\Gamma, PT(\Gamma, s)), \tilde{s}) \mid s \in DR(G),$ $\exists H \in s\ \exists \widetilde{H} \in \tilde{s}\ H \xrightarrow{\Gamma} \widetilde{H}\}$;*
- *the* initial state *is $s_G = [G]_\simeq$.*

A transition $(s, (\Gamma, \mathcal{P}), \tilde{s}) \in \mathcal{T}_G$ will be written as $s \xrightarrow{\Gamma}_{\mathcal{P}} \tilde{s}$. It is interpreted as follows: the probability to change the state s to $\tilde{s}$ in the result of executing Γ is $\mathcal{P}$. The step probabilities belong to the interval $(0;1]$. The value 1 is the case when we cannot leave a state, and thus there exists the only transition from the state to itself. We write $s \xrightarrow{\Gamma} \tilde{s}$ if $\exists \mathcal{P}\ s \xrightarrow{\Gamma}_{\mathcal{P}} \tilde{s}$ and $s \rightarrow \tilde{s}$ if $\exists \Gamma\ s \xrightarrow{\Gamma} \tilde{s}$.

Note that Γ could be the empty set, and its execution does not change isomorphism classes. This corresponds to the application of inaction rules to the expressions from the isomorphism classes. We have to keep track of such executions called *empty loops*, because they have nonzero probabilities. This follows from the definition of $PF(\emptyset, s)$ and the fact that multiaction probabilities cannot be equal to 1 as they belong to the interval $(0;1)$.

Definition 7. *Let G, G' be dynamic expressions, $TS(G) = (S_G, L_G, \mathcal{T}_G, s_G)$, $TS(G') = (S_{G'}, L_{G'}, \mathcal{T}_{G'}, s_{G'})$ be their transition systems. A mapping $\beta : S_G \rightarrow S_{G'}$ is an* isomorphism *between $TS(G)$ and $TS(G')$, denoted by $\beta : TS(G) \simeq TS(G')$, if β is a bijection such that $\beta(s_G) = s_{G'}$ and $\forall s, \tilde{s} \in S_G\ \forall \Gamma\ s \xrightarrow{\Gamma}_{\mathcal{P}} \tilde{s} \Leftrightarrow \beta(s) \xrightarrow{\Gamma}_{\mathcal{P}} \beta(\tilde{s})$. Two transition systems $TS(G)$ and $TS(G')$ are* isomorphic, *denoted by $TS(G) \simeq TS(G')$, if $\exists \beta : TS(G) \simeq TS(G')$.*

Definition 8. *Two dynamic expressions G and G' are* isomorphic w.r.t. transition systems, *denoted by $G =_{ts} G'$, if $TS(G) \simeq TS(G')$.*

Definition 9. *Let G be a dynamic expression. The* underlying discrete time Markov chain (DTMC) *of G, denoted by $DTMC(G)$, has the state space $DR(G)$ and the transitions $s \rightarrow_{\mathcal{P}} \tilde{s}$, if $s \rightarrow \tilde{s}$ and $\mathcal{P} = PM(s, \tilde{s})$.*

4 Denotational semantics

In this section, we construct the denotational semantics in terms of a subclass of LDTSPNs called discrete time stochastic Petri boxes (dts-boxes).

Definition 10. *A* plain discrete time stochastic Petri box (plain dts-box) *is a tuple $N = (P_N, T_N, W_N, \Lambda_N)$, where*

- *P_N and T_N are finite sets of* places *and* transitions*, respectively, with $P_N \cup T_N \neq \emptyset$ and $P_N \cap T_N = \emptyset$;*
- *$W_N : (P_N \times T_N) \cup (T_N \times P_N) \rightarrow I\!N$ is a function describing the* weights of arcs *between places and transitions and vice versa;*

– Λ_N *is the* place and transition labeling *function such that* $\Lambda_N : P_N \to \{\mathsf{e}, \mathsf{i}, \mathsf{x}\}$ *(it specifies* entry, internal *and* exit *places, respectively) and* $\Lambda_N : T_N \to \mathcal{SL}$ *(it associates activities with transitions).*

Let $t \in T_N,\ U \in I\!N_f^{T_N}$. *The* precondition ${}^\bullet t$ *and the* postcondition $t^\bullet$ *of* t *are the multisets of places defined as* $({}^\bullet t)(p) = W_N(p,t)$ *and* $(t^\bullet)(p) = W_N(t,p)$. *The* precondition ${}^\bullet U$ *and the* postcondition $U^\bullet$ *of* U *are the multisets of places defined as* ${}^\bullet U = \sum_{t \in U} {}^\bullet t$ *and* $U^\bullet = \sum_{t \in U} t^\bullet$. *We require that* $\forall t \in T_N\ {}^\bullet t \neq \emptyset \neq t^\bullet$. *In addition, for the set of* entry *places of* N *defined as* ${}^\circ N = \{p \in P_N \mid \Lambda_N(p) = \mathsf{e}\}$ *and the set of* exit *places of* N *defined as* $N^\circ = \{p \in P_N \mid \Lambda_N(p) = \mathsf{x}\}$ *we require that* ${}^\circ N \neq \emptyset \neq N^\circ$ *and* ${}^\bullet({}^\circ N) = \emptyset = (N^\circ)^\bullet$.

A *marked plain dts-box* is a pair (N, M_N), where N is a plain dts-box and $M_N \in I\!N_f^{P_N}$ is the *initial marking*. We denote $\overline{N} = (N, {}^\circ N)$ and $\underline{N} = (N, N^\circ)$. A marked plain dts-box $(P_N, T_N, W_N, \Lambda_N, M_N)$ can be interpreted as the LDTSPN $(P_N, T_N, W_N, \Omega_N, L_N, M_N)$, where $\forall t \in T_N\ \Omega_N(t) = \Omega(\Lambda_N(t))$ (*transition probability* function) and $L_N(t) = \mathcal{L}(\Lambda_N(t))$ (*transition labeling* function). The label τ of transitions of an LDTSPN corresponds to the multiaction part $\emptyset$ of activities that label transitions of the corresponding dts-box. The behaviour of marked dts-boxes follows from the firing rule of LDTSPNs.

To construct semantic function that associates a plain dts-box with every static expression of dtsPBC, we need to propose the *enumeration* function $Enu : T_N \to I\!N^*$ that associates the numbers with transitions of a plain dts-box N in accordance with the enumeration of activities of the underlying static expression. The structure of the plain dts-box corresponding to a static expression is constructed like in PBC, see [2]. I.e., we use simultaneous refinement and relabeling meta-operator (net refinement) in addition to the *operator dts-boxes* corresponding to the algebraic operations of dtsPBC and featuring transformational transition relabelings. In the definition of denotational semantics, we shall use standard constructions used for PBC. For convenience, we only use slightly different notation: φ, Θ and u stand for ρ (relabeling), Ω (operator box) and v (transition name) from PBC setting, respectively.

The relabeling relations $\varphi \subseteq I\!N_f^{\mathcal{SL}} \times \mathcal{SL}$ are defined as follows:

– $\varphi_{id} = \{(\{(\alpha,\rho)\}, (\alpha,\rho) \mid (\alpha,\rho) \in \mathcal{SL}\}$ is the *identity* relabeling;
– $\varphi_{[f]} = \{(\{(\alpha,\rho)\}, (f(\alpha),\rho) \mid (\alpha,\rho) \in \mathcal{SL}\}$;
– $\varphi_{\mathsf{rs}\ a} = \{(\{(\alpha,\rho)\}, (\alpha,\rho) \mid (\alpha,\rho) \in \mathcal{SL},\ a, \hat{a} \notin \mathcal{A}(\alpha)\}$;
– $\varphi_{\mathsf{sy}\ a}$ is the least relabeling relation in φ_{id} such that if $(\Gamma, \{(\alpha + \{a\}, \rho)\} \in \varphi_{\mathsf{sy}\ a}$ and $(\Delta, \{(\beta + \{\hat{a}\}, \chi)\} \in \varphi_{\mathsf{sy}\ a}$ then $(\Gamma + \Delta, \{(\alpha + \beta, \rho \cdot \chi)\} \in \varphi_{\mathsf{sy}\ a}$.

Now we define the enumeration function Enu. Let $Box_{dts}(E) = (P_E, T_E, W_E, \Omega_E, L_E)$ be the plain dts-box of a static expression E, and Enu_E be the enumeration function for T_E.

– $Box_{dts}(E \circ F) = \Theta_\circ(Box_{dts}(E), Box_{dts}(F)),\ \circ \in \{;, [], \|\}$. Since we do not introduce new transitions, we preserve the initial enumeration:
$$Enu(t) = \begin{cases} Enu_E(t),\ t \in T_E; \\ Enu_F(t),\ t \in T_F. \end{cases}$$

- $Box_{dts}(E[f]) = \Theta_{[f]}(Box_{dts}(E))$. Since we only change the labels of some multiactions by a bijection, we preserve the initial enumeration:
  $Enu(t) = Enu_E(t),\ t \in T_E$.
- $Box_{dts}(E \text{ rs } a) = \Theta_{\text{rs } a}(Box_{dts}(E))$. Since we remove all transitions labeled with multiactions containing a or $\hat{a}$, this does not change the enumeration of the remaining transitions: $Enu(t) = Enu_E(t),\ t \in T_E,\ a, \hat{a} \notin L_E(t)$.
- $Box_{dts}(E \text{ sy } a) = \Theta_{\text{sy } a}(Box_{dts}(E))$. Note that $\forall v, w \in T_E$ with $L_E(v) = \alpha + \{a\},\ L_E(w) = \beta + \{\hat{a}\}$, the new transition t resulting from synchronization of v and w has the label $L(t) = \alpha + \beta$, probability $\Omega(t) = \Omega_E(v) \cdot \Omega_E(w)$ and enumeration $Enu(t) = Enu_E(v) \cdot Enu_E(w)$. The enumeration is defined as
  $Enu(t) = \begin{cases} Enu_E(t), & t \in T_E; \\ Enu_E(v) \cdot Enu_E(w), & t \text{ results from synchronization of } v,\ w. \end{cases}$
- $Box_{dts}([E * F * K]) = \Theta_{[**]}(Box_{dts}(E), Box_{dts}(F), Box_{dts}(K))$. Since we do not introduce new transitions, we preserve the initial enumeration:
  $Enu(t) = \begin{cases} Enu_E(t),\ t \in T_E; \\ Enu_F(t),\ t \in T_F; \\ Enu_K(t),\ t \in T_K. \end{cases}$

Definition 11. *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 the area of plain dts-boxes defined as follows:*

1. $Box_{dts}((\alpha, \rho)_i) = N_{(\alpha,\rho)_i}$;
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. $Box_{dts}(E \circ a) = \Theta_{\circ a}(Box_{dts}(E)),\ \circ \in \{\text{rs}, \text{sy}\}$;
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 $Box_{dts}(\underline{E}) = \underline{Box_{dts}(E)}$. Note that any dynamic expression can be decomposed into overlined or underlined static expressions or those without overlines and underlines, and the definition of dts-boxes is compositional.

Isomorphism is a coincidence of systems up to renaming of their components or states. Let $\simeq$ denote isomorphism between transition systems or DTMCs and reachability graphs. Note that in this case, the names of transitions of the dts-box corresponding to a static expression could be identified with the enumerated activities of the latter. For a dts-box N, we denote its *reachability graph* by $RG(N)$ and its *underlying DTMC* by $DTMC(N)$.

Theorem 1. *[6] For any static expression E*

$$TS(\overline{E}) \simeq RG(Box_{dts}(\overline{E})).$$

Proposition 1. *[6] For any static expression E*

$$DTMC(\overline{E}) \simeq DTMC(Box_{dts}(\overline{E})).$$

5 Performance evaluation

Usually, stationary distribution is used for performance evaluation. Performance indices are then calculated based on the steady state probabilities.

5.1 Empty loops

To identify processes with intuitively similar behavior and to apply standard constructions and techniques, we should abstract from infinite internal behaviour.

Let G be a dynamic expression. A transition system $TS(G)$ can have loops going from a state to itself which are labeled by the empty set and have non-zero probability. Such *empty loop* $s \xrightarrow{\emptyset}_{\mathcal{P}} s$ appears when no activities occur at a time step, and this happens with some positive probability. Obviously, in this case the current state remains unchanged.

Let G be a dynamic expression and $s \in DR(G)$. The *probability to stay in* s *due to* k $(k \geq 1)$ *empty loops* is $(PT(\emptyset, s))^k$. The *probability to execute in* s *a non-empty multiset of activities* $\Gamma \in Exec(s) \setminus \{\emptyset\}$ *after possible empty loops* is

$$PT^*(\Gamma, s) = PT(\Gamma, s) \cdot \sum_{k=0}^{\infty} (PT(\emptyset, s))^k = \frac{PT(\Gamma, s)}{1 - PT(\emptyset, s)}.$$

The value $k = 0$ in the summation above corresponds to the case when no empty loops occur. Note that $PT^*(\Gamma, s) \leq 1$, hence, it is really a probability, since $PT(\emptyset, s) + PT(\Gamma, s) \leq PT(\emptyset, s) + \sum_{\Delta \in Exec(s)\setminus\{\emptyset\}} PT(\Delta, s) = \sum_{\Delta \in Exec(s)} PT(\Delta, s) = 1$. Moreover, $PT^*(\Gamma, s)$ defines a probability distribution, i.e., $\forall s \in DR(G)\ \sum_{\Gamma \in Exec(s)\setminus\{\emptyset\}} PT^*(\Gamma, s) = 1$.

Definition 12. *The* (labeled probabilistic) transition system without empty loops $TS^*(G)$ *has the state space* $DR(G)$ *and the transitions* $s \overset{\Gamma}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s}$, *if* $s \xrightarrow{\Gamma} \tilde{s}$, $\Gamma \neq \emptyset$ *and* $\mathcal{P} = PT^*(\Gamma, s)$.

We write $s \overset{\Gamma}{\twoheadrightarrow} \tilde{s}$ if $\exists \mathcal{P}\ s \overset{\Gamma}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s}$ and $s \twoheadrightarrow \tilde{s}$ if $\exists \Gamma\ s \overset{\Gamma}{\twoheadrightarrow} \tilde{s}$.

Definition 13. *Two dynamic expressions* G *and* G' *are* isomorphic w.r.t. transition systems without empty loops, *denoted by* $G =_{ts*} G'$, *if* $TS^*(G) \simeq TS^*(G')$.

Definition 14. *The* underlying DTMC without empty loops $DTMC^*(G)$ *has the state space* $DR(G)$ *and the transitions* $s \twoheadrightarrow_{\mathcal{P}} \tilde{s}$, *if* $s \twoheadrightarrow \tilde{s}$, *where* $\mathcal{P} = PM^*(s, \tilde{s})$ *and*

$$PM^*(s, \tilde{s}) = \sum_{\{\Gamma \mid s \overset{\Gamma}{\twoheadrightarrow} \tilde{s}\}} PT^*(\Gamma, s).$$

Let $N = (P_N, T_N, W_N, \Omega_N, L_N, M_N)$ be a LDTSPN and $M, \widetilde{M} \in I\!N_f^{P_N}$, $t \in T_N$, $U \subseteq T_N$. Then the transition relations $M \overset{U}{\twoheadrightarrow}_{\mathcal{P}} \widetilde{M}$, $M \overset{U}{\twoheadrightarrow} \widetilde{M}$, $M \twoheadrightarrow \widetilde{M}$, $M \twoheadrightarrow_{\mathcal{P}} \widetilde{M}$, the *reachability graph without empty loops* $RG^*(N)$ and the *underlying DTMC without empty loops* $DTMC^*(N)$ are defined like the corresponding notions for dynamic expressions.

Theorem 2. *For any static expression E*

$$TS^*(\overline{E}) \simeq RG^*(Box_{dts}(\overline{E})).$$

Proof. For the qualitative behaviour, we have the same isomorphism as in PBC. The quantitative behaviour is the same, since the activities of an expression have probability parts coinciding with the probabilities of the transitions belonging to the corresponding dts-box and, both in stochastic processes specified by expressions and dts-boxes conflicts are resolved via the same probability functions. ☐

Proposition 2. *For any static expression E*

$$DTMC^*(\overline{E}) \simeq DTMC^*(Box_{dts}(\overline{E})).$$

Proof. By Theorem 2 and definitions of underlying DTMC for dynamic expressions and LDTSPNs, since transition probabilities of the associated DTMCs are the sums of those belonging to transition systems or reachability graphs. ☐

Theorem 2 guarantees that the net versions of algebraic equivalences could be easily defined. For every equivalence on the transition system without empty loops of a dynamic expression, a similarly defined analogue exists on the reachability graph without empty loops of the corresponding dts-box.

5.2 Stationary behaviour

Let us describe stationary behaviour of infinite stochastic processes specified by expressions of dtsPBC. We shall consider only formulas specifying stochastic processes with an infinite behaviour, thus, the expressions with iteration operator. We suppose that the underlined DTMC of each such an expression is irreducible or contains at least only one irreducible subset of states to guarantee an existence of the steady state.

Let G be a dynamic expression. The elements $\mathcal{P}^*_{ij}$ $(1 \le i, j \le n = |DR(G)|)$ of transition probability matrix (TPM) $\mathbf{P}^*$ for $DTMC^*(G)$ are defined as

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

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 the equation system

$$\psi^*[k] = \psi^*[0](\mathbf{P}^*)^k,$$

where $\psi^*[0] = (\psi^*_1[0], \ldots, \psi^*_n[0])$ is the initial PMF defined as $\psi^*_i[0] = \begin{cases} 1, & s_i = [G]_\simeq; \\ 0, & \text{otherwise.} \end{cases}$ Note also that $\psi^*[k+1] = \psi^*[k]\mathbf{P}^*$ $(k \in \mathbb{N})$.

The steady state PMF $\psi^* = (\psi^*_1, \ldots, \psi^*_n)$ for $DTMC^*(G)$ is the solution of the equation system

$$\begin{cases} \psi^*(\mathbf{P}^* - \mathbf{E}) = \mathbf{0} \\ \psi^*\mathbf{1}^T = 1 \end{cases},$$

where $\mathbf{E}$ is the unitary matrix of size n and $\mathbf{0}$ is a vector with n values 0, $\mathbf{1}$ is that with n values 1.

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

5.3 Shared memory system

Consider a model of two processors accessing a common shared memory described in [3] in the continuous time setting on GSPNs. We analyze this shared memory system in the discrete time setting within dtsPBC where concurrent execution of activities is possible. The model performs as follows. After activation of the system (turning the computer on), 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 should wait until the former one ends its memory operations, and the system returns to the state with both active processors and the available common memory.

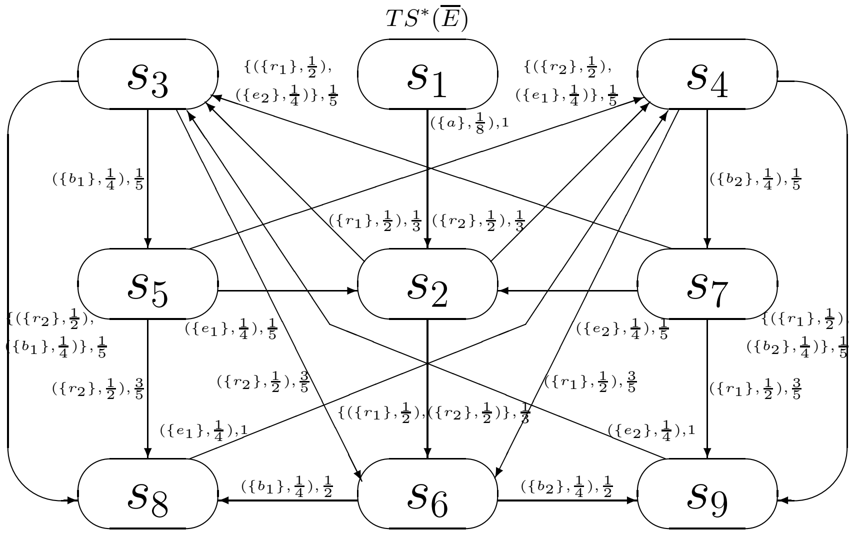
The meaning of actions from expressions specifying the system modules is as follows. The action a corresponds to the system activation. The actions r_i $(1 \le i \le 2)$ represent the common memory request of processor i. The actions b_i and e_i correspond to the beginning and the end, respectively, of the common memory access of processor i. The other actions are used for communication purpose only via synchronization, and we abstract from them later using restriction. The expression $\mathsf{Stop} = (\{c\}, \frac{1}{2})$ rs c specifies the process that is only able to perform empty loops with probability 1 and never terminates.

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. $DR(\overline{E})$ consists of 9 isomorphism classes which are not presented here to avoid long and complex notation.

In Figure 1 the transition system without empty loops $TS^*(\overline{E})$ is presented.

The TPM for $DTMC^*(\overline{E})$ is $\mathbf{P}^* = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{5} & \frac{3}{5} & 0 & \frac{1}{5} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{3}{5} & \frac{1}{5} & 0 & \frac{1}{5} \\ 0 & \frac{1}{5} & 0 & \frac{1}{5} & 0 & 0 & 0 & \frac{3}{5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{5} & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 & \frac{3}{5} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$.

In Table 1 the values are presented, and in Figure 2 an alteration diagram is depicted for the transient state probabilities $\psi_i^*[k]$ $(i \in \{1, 2, 3, 5, 6, 8\})$ of the shared memory system at the time moments k $(0 \le k \le 10)$. It is sufficient to consider 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 as well as for s_8, s_9.



$TS^*(\overline{E})$
s_1
s_2
s_3
s_4
s_5
s_6
s_7
s_8
s_9
$(\{a\},\frac{1}{8}),1$
$\{(\{r_1\},\frac{1}{2}),(\{e_2\},\frac{1}{4})\},\frac{1}{5}$
$\{(\{r_2\},\frac{1}{2}),(\{e_1\},\frac{1}{4})\},\frac{1}{5}$
$(\{b_1\},\frac{1}{4}),\frac{1}{5}$
$(\{b_2\},\frac{1}{4}),\frac{1}{5}$
$(\{r_1\},\frac{1}{2}),\frac{1}{3}$
$(\{r_2\},\frac{1}{2}),\frac{1}{3}$
$\{(\{r_2\},\frac{1}{2}),(\{b_1\},\frac{1}{4})\},\frac{1}{5}$
$(\{e_1\},\frac{1}{4}),\frac{1}{5}$
$(\{e_2\},\frac{1}{4}),\frac{1}{5}$
$\{(\{r_1\},\frac{1}{2}),(\{b_2\},\frac{1}{4})\},\frac{1}{5}$
$(\{r_2\},\frac{1}{2}),\frac{3}{5}$
$(\{r_2\},\frac{1}{2}),\frac{3}{5}$
$(\{r_1\},\frac{1}{2}),\frac{3}{5}$
$(\{r_1\},\frac{1}{2}),\frac{3}{5}$
$\{(\{r_1\},\frac{1}{2}),(\{r_2\},\frac{1}{2})\},\frac{1}{3}$
$(\{e_1\},\frac{1}{4}),1$
$(\{e_2\},\frac{1}{4}),1$
$(\{b_1\},\frac{1}{4}),\frac{1}{2}$
$(\{b_2\},\frac{1}{4}),\frac{1}{2}$

Fig. 1. The transition system without empty loops of the shared memory system

Table 1. Transient state probabilities of the shared memory system

k	0	1	2	3	4	5	6	7	8	9	10	∞
$\psi_1^*[k]$	1	0	0	0	0	0	0	0	0	0	0	0
$\psi_2^*[k]$	0	1	0	0	0.0267	0	0.0197	0.0199	0.0047	0.0199	0.0160	0.0144
$\psi_3^*[k]$	0	0	0.3333	0	0.2467	0.2489	0.0592	0.2484	0.2000	0.1071	0.2368	0.1794
$\psi_5^*[k]$	0	0	0	0.0667	0	0.0493	0.0498	0.0118	0.0497	0.0400	0.0214	0.0359
$\psi_6^*[k]$	0	0	0.3333	0.4000	0	0.3049	0.2987	0.0776	0.3047	0.2416	0.1351	0.2201
$\psi_8^*[k]$	0	0	0	0.2333	0.2400	0.0493	0.2318	0.1910	0.0956	0.2221	0.1662	0.1675

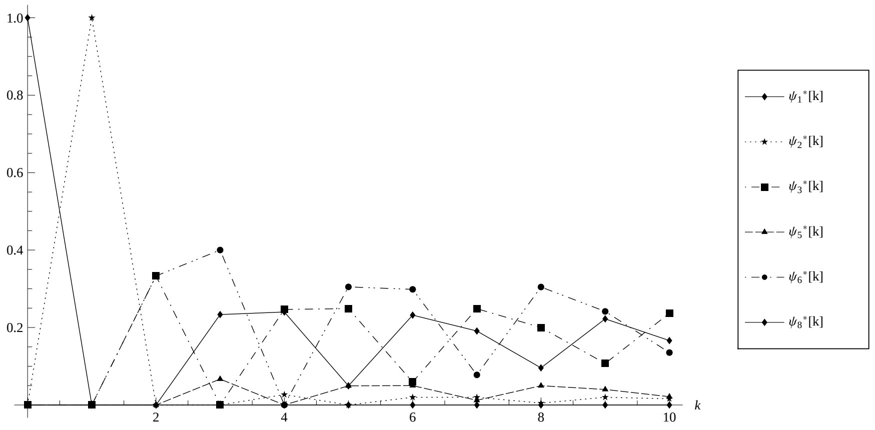


Fig. 2. Transient state probabilities alteration diagram of the shared memory system

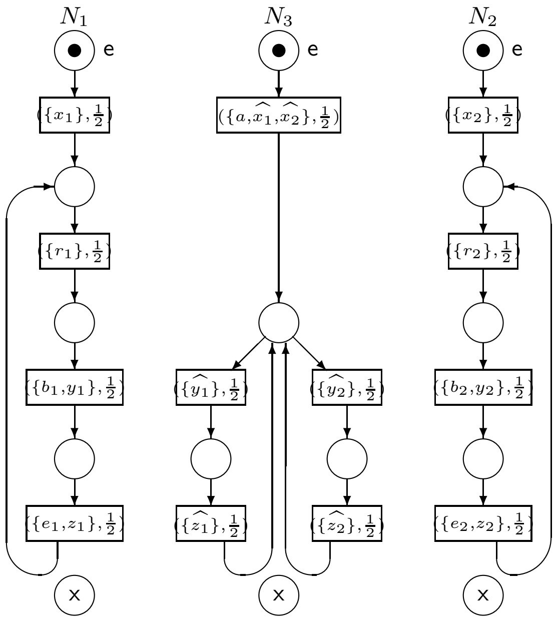
The steady state PMF ψ^* for $DTMC^*(\overline{E})$ is

$$\psi^* = \left(0, \frac{3}{209}, \frac{75}{418}, \frac{75}{418}, \frac{15}{418}, \frac{46}{209}, \frac{15}{418}, \frac{35}{209}, \frac{35}{209}\right).$$

We can now calculate performance indices.

- The average recurrence time in the state s_2, where no processor requests the memory, called the *average system run-through*, is $\frac{1}{\psi_2^*} = \frac{209}{3} = 69\frac{2}{3}$.
- The common memory is available in the states s_2, s_3, s_4, s_6. The steady state probability that the memory is available is $\psi_2^* + \psi_3^* + \psi_4^* + \psi_6^* = \frac{3}{209} + \frac{75}{418} + \frac{75}{418} + \frac{46}{209} = \frac{124}{209}$. Then the steady state probability that the memory is used (i.e., not available) called the *shared memory utilization* is $1 - \frac{124}{209} = \frac{85}{209}$.
- The common memory request of the first processor $(\{r_1\}, \frac{1}{2})$ is 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 $(\{r_1\}, \frac{1}{2})$. Thus, the *steady state probability of the shared memory request from the first processor* is $\psi_2^* \sum_{\{\Gamma | (\{r_1\}, \frac{1}{2}) \in \Gamma\}} PT^*(\Gamma, s_2) + \psi_4^* \sum_{\{\Gamma | (\{r_1\}, \frac{1}{2}) \in \Gamma\}} PT^*(\Gamma, s_4) + \psi_7^* \sum_{\{\Gamma | (\{r_1\}, \frac{1}{2}) \in \Gamma\}} PT^*(\Gamma, s_7) = \frac{3}{209} \cdot (\frac{1}{3} + \frac{1}{3}) + \frac{75}{418} \cdot (\frac{3}{5} + \frac{1}{5}) + \frac{15}{418} \cdot (\frac{3}{5} + \frac{1}{5}) = \frac{38}{209}$.

In Figure 3 the marked dts-boxes corresponding to the dynamic expressions of two processors and shared memory are depicted, i.e., $N_i = Box_{dts}(\overline{E_i})$ $(1 \leq i \leq 3)$. In Figure 4 the marked dts-box corresponding to the dynamic expression of the shared memory system is presented, i.e., $N = Box_{dts}(\overline{E})$.


N_1
N_3
N_2
e
e
e
$\{x_1\}, \frac{1}{2}$
$(\{a, \widehat{x_1}, \widehat{x_2}\}, \frac{1}{2})$
$\{x_2\}, \frac{1}{2}$
$\{r_1\}, \frac{1}{2}$
$\{r_2\}, \frac{1}{2}$
$\{b_1, y_1\}, \frac{1}{2}$
$\{\widehat{y_1}\}, \frac{1}{2}$
$\{\widehat{y_2}\}, \frac{1}{2}$
$\{b_2, y_2\}, \frac{1}{2}$
$\{e_1, z_1\}, \frac{1}{2}$
$\{\widehat{z_1}\}, \frac{1}{2}$
$\{\widehat{z_2}\}, \frac{1}{2}$
$\{e_2, z_2\}, \frac{1}{2}$
x
x
x
Fig. 3. The marked dts-boxes of two processors and shared memory

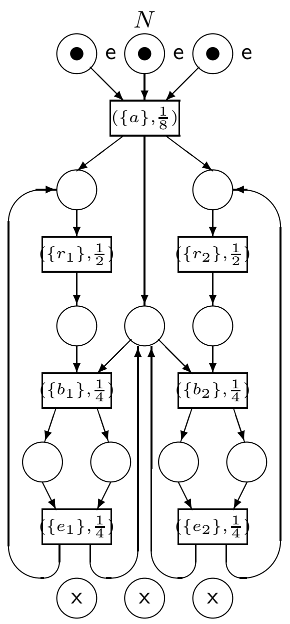


Fig. 4. The marked dts-box of the shared memory system

6 Conclusion

In this paper, within dtsPBC with iteration, a method of performance evaluation of concurrent stochastic systems was proposed based on steady state probabilities analysis and applied to the shared memory system.

We plan to define and investigate stochastic equivalences of dtsPBC which allow one to identify stochastic processes with similar behaviour that are differentiated by too strict notion of the semantic equivalence. Moreover, we would like to extend dtsPBC with recursion to enhance specification power of the calculus.

References

1. Best E., Devillers R., Hall J.G. *The box calculus: a new causal algebra with multi-label communication. Lect. Notes Comp. Sci.* **609**, p. 21–69, 1992.
2. Best E., Koutny M. *A refined view of the box algebra. Lect. Notes Comp. Sci.* **935**, p. 1–20, 1995.
3. Marsan M.A., Balbo G., Conte G., Donatelli S., Franceschinis G. *Modelling with generalized stochastic Petri nets.* John Wiley and Sons, 316 p., 1995.
4. Macià H.S., Valero V.R., Cazorla D.L., Cuartero F.G. *Introducing the iteration in sPBC. Lect. Notes Comp. Sci.* **3235**, p. 292–308, 2004.
5. 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 PNPM'01*, p. 207–216, Aachen, Germany, IEEE Computer Society Press, 2001.
6. Tarasyuk I.V. *Iteration in discrete time stochastic Petri box calculus. Bulletin of the NCC, Comp. Sci., IIS Issue* **24**, p. 129–148, NCC Publisher, Novosibirsk, 2006.
7. Tarasyuk I.V. *Stochastic Petri box calculus with discrete time. Fundamenta Informaticae* **76(1–2)**, p. 189–218, IOS Press, Amsterdam, The Netherlands, 2007.