Under consideration for publication in Math. Struct. in Comp. Science

Equivalence relations for modular performance evaluation in dtsPBC †

IGOR V. TARASYUK

A.P. Ershov Institute of Informatics Systems,
Siberian Branch of the Russian Academy of Sciences,
6, Acad. Lavrentiev ave., 630090 Novosibirsk, Russian Federation.
E-mail: itar@iis.nsk.su

Received 15 September 2011; Revised 9 January 2013

In the framework of a discrete time stochastic extension dtsPBC of finite Petri box calculus (PBC) enriched with iteration, we define a number of stochastic equivalences. They allow one to identify stochastic processes with similar behaviour that are however differentiated by the semantics of the calculus. We explain in which way the equivalences we propose can be used to reduce transition systems of expressions. It is demonstrated how to apply the equivalences to compare the stationary behaviour. The equivalences guarantee a coincidence of performance indices for stochastic systems and can be used for performance analysis simplification. In a case study, a method of modeling, performance evaluation and behaviour preserving reduction of concurrent computing systems is outlined and applied to the dining philosophers system.

Keywords: stochastic process algebra, Petri box calculus, iteration, discrete time, stochastic equivalence, reduction, stationary behaviour, performance evaluation.

1. Introduction

Process algebras (PAs), such as CCS (Milner 1989), are a widely used formal model designed to specify concurrent systems and analyze their behavioural properties. In such calculi, processes are specified by compositional formulas constructed by operators from symbols of actions, and verification of properties is accomplished syntactically by means of algebraic laws and equivalences. In the last decades, stochastic extensions of PAs were proposed. Stochastic process algebras (SPAs) do not just specify actions which can occur (qualitative features) like standard PAs, but they associate with actions the distribution parameters of their random time delays (quantitative characteristics). The best-known SPAs are MTIPP (Hermanns and Rettelbach 1994), PEPA (Hillston 1996) and EMPA (Bernardo and Gorrieri 1998).

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

1.1. *Petri box calculus*

PAs specify concurrent systems in a compositional way via an expressive formal syntax. On the other hand, Petri nets (PNs) provide a graphical representation of such systems and capture explicit asynchrony in their behaviour. To combine advantages of both models, a semantics of algebraic formulas in terms of PNs is defined. Petri box calculus (PBC) (Best *et al.* 1992; Best and Koutny 1995) is a flexible and expressive process algebra intended to provide support for compositional translation from high level concurrent programming languages into PNs. Formulas of PBC are combined not from single actions, like in CCS, but from multisets of elementary actions and their conjugates, called multiactions (*basic formulas*). In contrast to CCS, synchronization is separated from parallelism (*concurrent constructs*). Synchronization is defined as a unary multi-way stepwise operation based on communication of actions and their conjugates, thus, it extends the CCS approach with conjugate matching labels. Synchronization in PBC is asynchronous, unlike that in Synchronous CCS (SCCS) (Milner 1989). The other operations are sequence and choice (*sequential constructs*). The calculus includes also restriction and relabeling (*abstraction constructs*). To specify infinite processes, the refinement, recursion and iteration operations were added (*hierarchical constructs*). Thus, unlike CCS, PBC has an additional iteration construction to specify infinite behaviour when the semantic interpretation in finite PNs is possible. PBC has a step operational semantics in terms of labeled transition systems based on rules in the Structured Operational Semantics (SOS) style. A denotational semantics of PBC was proposed via a subclass of PNs equipped with an interface and considered up to isomorphism, called Petri boxes. Recently, the extensions of PBC with deterministic, a nondeterministic or stochastic time were presented.

1.2. *Time extensions of Petri box calculus*

To specify systems with time constraints, such as real time systems, deterministic (fixed) or nondeterministic (interval) time delays are used. A time extension of PBC with a nondeterministic time model, called time Petri box calculus (tPBC), was proposed in (Koutny 2000). In tPBC, timing information is added by associating time intervals (the earliest and the latest firing time) with instantaneous *actions*. Its denotational semantics was defined in terms of a subclass of labeled time Petri nets (LtPNs), based on tPNs (Merlin and Farber 1976), and called time Petri boxes (ct-boxes). tPBC has a step time operational semantics in terms of labeled transition systems. Another time enrichment of PBC, called timed Petri box calculus (TPBC), was defined in (Marroquín and de-Frutos 2001), it accommodates a deterministic model of time. In contrast to tPBC, multiactions of TPBC are not instantaneous, but have time durations. Additionally, in TPBC there exist no “illegal” multiaction occurrences, unlike tPBC. The complexity of “illegal” occurrences mechanism was one of the main intentions to construct TPBC though this calculus appeared to be more complicated than tPBC. The denotational semantics of TPBC was defined in terms of a subclass of labeled timed Petri nets (LTPNs), based on TPNs (Ramchandani 1973), and called timed Petri boxes (T-boxes). TPBC has a step timed operational semantics in terms of labeled transition systems. Note that tPBC and TPBC differ in ways they capture time information, and they are not in competition but complement each other. The third time extension of PBC, called arc time Petri box calculus (atPBC), was constructed in (Niaouris 2005), and it implements a nondeterministic time. In atPBC, multiactions are associated with time delay intervals. Its denotational semantics was defined on a subclass of labeled arc time Petri nets (atPNs), where time restrictions are associated with the arcs, called arc time Petri boxes (at-boxes). atPBC possesses a step operational semantics in terms of labeled transition systems.

The set of states for the systems with deterministic or nondeterministic delays differs drastically from that for the untime systems, hence, the analysis results for untime systems may be not valid for the time ones. To solve this problem, stochastic delays are considered, which are the random values with a (discrete or continuous) probability distribution. A continuous time stochastic extension of a finite part of PBC called stochastic Petri box calculus (sPBC) was proposed in (Macià *et al.* 2001). sPBC in its former version had neither refinement nor recursion nor iteration operations and thus specified finite processes only. An interleaving operational semantics of the calculus was constructed via labeled probabilistic transition systems. A denotational semantics of sPBC was defined in terms of a subclass of labeled continuous time stochastic PNs (LCTSPNs), based on CTSPNs (Marsan 1990), and called stochastic Petri boxes (s-boxes). In (Macià *et al.* 2004), the iteration operation was added to sPBC to specify infinite processes and the producer/consumer system was specified. In sPBC with iteration, performance of the processes is evaluated by analyzing their underlying continuous time Markov chains (CTMCs). In (Macià *et al.* 2008), the resulting calculus was enriched with immediate multiactions, and a manufacturing system, as well as the AUY-protocol, were modeled. A denotational semantics of such an sPBC extension (we call it generalized sPBC or gsPBC) was defined via a subclass of labeled generalized SPNs (LGSPNs), based on GSPNs (Marsan 1990), and called generalized stochastic Petri boxes (gs-boxes). The performance analysis in gsPBC is based on the underlying semi-Markov chains (SMCs). The example systems considered within sPBC and its extensions had an interleaving semantics. The performance indices were calculated only for the systems from (Macià *et al.* 2008).

PBC has a step operational semantics, whereas sPBC has only an interleaving one, hence, a stochastic extension of PBC with a step semantics is needed to keep the concurrency degree of behavioural analysis at the same level as in PBC. A discrete time stochastic extension dtsPBC of finite PBC was presented in (Tarasyuk 2005; Tarasyuk 2007). A step operational semantics of the algebra was constructed with the use of labeled probabilistic transition systems. dtsPBC has a denotational semantics in terms of a subclass of labeled discrete time stochastic PNs (LDTSPNs), based on DTSPNs (Molloy 1985), and called discrete time stochastic Petri boxes (dts-boxes). A number of stochastic equivalences were proposed to identify stochastic processes with similar behaviour which are differentiated by the semantic equivalence. The interrelations of the introduced equivalences were studied. In (Tarasyuk 2006; Tarasyuk 2008), the syntax of dtsPBC was supplemented by the iteration operator. The performance evaluation in dtsPBC with iteration is accomplished via the underlying discrete time Markov chains

(DTMCs) of the algebraic processes. In (Tarasyuk *et al.* 2010), we presented the extension dtsiPBC of the latter calculus with immediate multiactions. The performance analysis in dtsiPBC is based on the underlying semi-Markov chains (SMCs) and (reduced) DTMCs.

Since dtsPBC has a discrete time semantics and geometrically distributed sojourn time in the process states, unlike sPBC with continuous time semantics and exponentially distributed delays, the calculi apply two different approaches to the stochastic extension of PBC, in spite of some similarity of their syntax and semantics inherited from PBC. The main advantage of dtsPBC is that concurrency is treated naturally, like in PBC, whereas in sPBC parallelism is simulated by interleaving, obliging one to collect the information on causal independence of activities before constructing the semantics.

If to compare dtsPBC with classical SPAs MTIPP, PEPA, EMPA, the first main difference between them comes from PBC, since dtsPBC is based on this calculus: all algebraic operations and a notion of multiaction are inherited from PBC. The second main difference is discrete probabilities of activities induced by the discrete time approach, whereas action rates are used in the standard SPAs with continuous time. As a consequence, dtsPBC has a non-interleaving step operational semantics. This is in contrast to the classical SPAs, where concurrency is modeled by interleaving because of the continuous probability distributions of action delays and the race condition applied when several actions can be executed in a state. The salient point of dtsPBC is a combination of discrete stochastic time and step semantics in an SPA.

1.3. *Equivalence relations*

A notion of equivalence is important in theory of computing systems. Equivalences are applied both to compare behaviour of systems and reduce their structure. There is a wide diversity of behavioural equivalences, and their interrelations were well explored in the literature. The most well-known and widely used one is bisimulation. Typically, the mentioned equivalences take into account only functional (qualitative) but not performance (quantitative) aspects. Additionally, the equivalences are usually interleaving ones, i.e. they interpret concurrency as sequential nondeterminism. To respect quantitative features of behaviour, probabilistic equivalences have additional requirement on execution probabilities. Two equivalent processes must be able to execute the same sequences of actions, and for every such sequence, its execution probabilities within both processes should coincide. In case of bisimulation equivalence, the states from which similar future behaviours start are grouped into equivalence classes that form elements of the aggregated state space. From every two bisimilar states, the same actions can be executed, and the subsequent states resulting from execution of an action belong to the same equivalence class. In addition, for both states, the cumulative probabilities to move to the same equivalence class by executing the same action coincide. A different kind of quantitative relations are Markovian equivalences, which take rate (the parameter of exponential distribution that governs time delays) instead of probability.

Interleaving probabilistic strong bisimulation equivalence was proposed in (Larsen and Skou 1991) on labeled probabilistic transition systems. Interleaving Markovian strong bisimulation equivalence was constructed in (Hermanns and Rettelbach 1994) for MTIPP, in (Hillston 1996) for PEPA and in (Bernardo and Gorrieri 1998) for EMPA. Interleaving probabilistic equivalences were defined for probabilistic processes in (Jou and Smolka 1990; van Glabbeek *et al.* 1995). Interleaving Markovian weak bisimulation equivalence was introduced in (Buchholz 1995) on labeled CTSPNs and in (Buchholz 1998) on generalized SPNs (GSPNs). In (Bernardo 2007), a comparison of a variety of interleaving Markovian trace, test and bisimulation equivalences was carried out on sequential and concurrent Markovian process calculi. At the same time, no appropriate equivalence notion was defined for concurrent SPAs so far.

1.4. *Contributions of the paper*

In this paper, a problem of performance preservation by the equivalence notions is discussed within dtsPBC enriched with iteration. First, we present the syntax of the calculus. Second, we describe its operational semantics in terms of labeled transition systems and its denotational semantics based on a subclass of LDTSPNs. Further, we propose a number of stochastic equivalences. We describe how the stochastic equivalences can be used to reduce transition systems of expressions and the related formalisms while preserving their qualitative and quantitative behaviour. We investigate which equivalences guarantee identity of the stationary behaviour. The mentioned property implies a coincidence of performance indices based on steady-state probabilities of modeled stochastic systems. The equivalences possessing the property can be used to reduce the state space of a system and thus simplify its performance evaluation, that is usually complex due to the state space explosion problem. At the end, we present a case study of the dining philosophers system explaining how to model concurrent computing systems within the calculus and analyze their performance as well as in which way to reduce the systems preserving their performance indices and making simpler the performance evaluation. Thus, the main contributions of the paper are the following.

— New powerful and expressive discrete time SPA dtsPBC.
— Step operational semantics of dtsPBC via labeled probabilistic transition systems.
— Petri net denotational semantics of dtsPBC via discrete time stochastic Petri nets.
— Performance analysis based on underlying discrete time Markov chains.
— Stochastic equivalence used for reduction that simplifies the performance evaluation.
— Extended case study illustrating how to apply the theoretical results in practice.

1.5. *Structure of the paper*

The paper is organized as follows. The syntax of dtsPBC is presented in Section 2. Section 3 describes the operational semantics of the calculus and Section 4 presents its denotational semantics. Stochastic algebraic equivalences are defined and investigated in Section 5. In Section 6 we explain how to reduce transition systems and the related formalisms modulo the equivalences. Section 7 is devoted to the application of the relations to the stationary behaviour comparison and determining the performance preserving equivalences. Section 8 describes specification, performance evaluation and reduction of the dining philosophers system within the calculus. The difference between dtsPBC and other well-known or similar SPAs is considered in Section 9. The advantages of dtsPBC are explained in Section 10. The concluding Section 11 summarizes the results obtained and outlines research perspectives in this area.

2. Syntax

In this section, we propose the syntax of the discrete time stochastic extension of finite PBC enriched with iteration, *discrete time stochastic* PBC (dtsPBC).

Definition 2.1. Let X be a set. A finite *multiset (bag)* M over X is a mapping $M : X \to I\!N$ such that $|\{x \in X \mid M(x) > 0\}| < \infty$, i.e. it can contain a finite number of elements only.

We denote the *set of all finite multisets* over a set X by $I\!N_f^X$. The *cardinality* of a multiset M is defined as $|M| = \sum_{x\in X} M(x)$. We write $x \in M$ if $M(x) > 0$ and $M \subseteq M'$ if for all $x \in X$ we have $M(x) \leq M'(x)$. We define $(M + M')(x) = M(x) + M'(x)$ and $(M - M')(x) = \max\{0, M(x) - M'(x)\}$. When for all $x \in X$ we have $M(x) \leq 1$ then M can be interpreted as a proper set and denoted by $M \subseteq X$.

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 $\hat{a} \neq a$ and $\hat{\hat{a}} = a$. Let $\mathcal{A} = Act \cup \widehat{Act}$ be the set of *all actions*, and $\mathcal{L} = I\!N_f^{\mathcal{A}}$ be the set of *all multiactions*. Note that $\emptyset \in \mathcal{L}$, this corresponds to an internal move, i.e. 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. This probability is interpreted as that of independent execution of the stochastic multiaction at the next discrete time moment. Such probabilities are used to calculate those to execute (possibly empty) multisets of stochastic multiactions after one time unit delay. The probabilities of stochastic multiactions are required not to be equal to 1 to avoid extra model complexity due to assigning with them weights needed to make a choice when several stochastic multiactions with probability 1 can be executed from a state. In this case, some problems appear with conflicts resolving. See (Molloy 1985) for the discussion on SPNs. This decision also allows us to avoid technical difficulties related to conditioning events with probability 0. Another reason is that not allowing probability 1 for multiactions excludes a potential source of periodicity (hence, non-ergodicity) in the underlying DTMCs of the algebraic expressions. On the other hand, there is no sense to allow zero probabilities of multiactions, since they would never be performed in this case. 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 $\mathcal{A}(\alpha, \rho) = \mathcal{A}(\alpha)$. The *alphabet* of $\Gamma \in I\!N_f^{\mathcal{SL}}$ is defined as $\mathcal{A}(\Gamma) = \cup_{(\alpha,\rho)\in\Gamma}\mathcal{A}(\alpha)$. For $(\alpha, \rho) \in \mathcal{SL}$, we define its *multiaction part* as $\mathcal{L}(\alpha, \rho) = \alpha$ and its *probability part* as $\Omega(\alpha, \rho) = \rho$. The *multiaction part* of $\Gamma \in I\!N_f^{\mathcal{SL}}$ is defined as $\mathcal{L}(\Gamma) = \sum_{(\alpha,\rho)\in\Gamma} \alpha$. Remember that sums and products are considered with the multiplicity when applied to multisets.

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.

Sequential execution and choice have a standard interpretation, like in other process algebras, but parallelism does not include synchronization, unlike the corresponding operation in *CCS*.

Relabeling functions $f : \mathcal{A} \to \mathcal{A}$ are bijections preserving conjugates, i.e. for all $x \in \mathcal{A}$ we have $f(\hat{x}) = \widehat{f(x)}$. Relabeling is extended to multiactions in a usual way: for $\alpha \in \mathcal{L}$ we define $f(\alpha) = \sum_{x\in\alpha} f(x)$. Relabeling is extended to the multisets of activities as follows: for $\Gamma \in I\!N_f^{\mathcal{SL}}$ we define $f(\Gamma) = \sum_{(\alpha,\rho)\in\Gamma}(f(\alpha), \rho)$.

Restriction over an elementary action $a \in Act$ means that, for a given expression, any process behaviour containing a or its conjugate $\hat{a}$ is not allowed.

Let $\alpha, \beta \in \mathcal{L}$ be two multiactions such that for some elementary 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 $\gamma(x) = \begin{cases} \alpha(x) + \beta(x) - 1, & \text{if } x = a \text{ or } x = \hat{a}; \\ \alpha(x) + \beta(x), & \text{otherwise.} \end{cases}$ In other words, we require that $\alpha \oplus_a \beta = \alpha + \beta - \{a, \hat{a}\}$, i.e. we remove one exemplar of a and one exemplar of $\hat{a}$ from the multiset sum $\alpha + \beta$, since the synchronization of a and $\hat{a}$ produces $\emptyset$.

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

Static expressions specify the structure of processes. The expressions correspond to unmarked LDTSPNs (note that LDTSPNs are marked by definition). Remember that a marking is the allocation of tokens in the places of a PN and markings are used to describe dynamic behaviour of PNs in terms of transition firings.

Definition 2.2. Let $(\alpha, \rho) \in \mathcal{SL},\ a \in Act$. A *static expression* of dtsPBC is

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

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: $(E; E),\ (E[]E),\ (E\|E)$ or to associate priorities with operations. However, we prefer the PBC approach, i.e. we add parentheses to resolve ambiguities and we assume no priorities.

To avoid technical difficulties with the iteration operator, we should not allow any concurrency at the highest level of the second argument of iteration. This is not a severe restriction though, since we can always prefix parallel expressions by an activity with the empty multiaction. Later on, in Example 4.3, we shall demonstrate that relaxing the restriction can result in nets which are not safe. Alternatively, we can use a different, safe, version of the iteration operator, but its net translation has six arguments. See also (Best *et al.* 2001) for discussion on this subject. Remember that a PN is *n-bounded* ($n \in I\!N$) if for all its reachable (from the initial marking by the sequences of transition firings) markings there are at most n tokens in every place, and a PN is *safe* if it is 1-bounded.

Definition 2.3. Let $(\alpha, \rho) \in \mathcal{SL},\ a \in Act$. A *regular static expression* of dtsPBC is

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

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

Dynamic expressions specify the states of processes. The expressions correspond to LDTSPNs (marked by default). Dynamic expressions are obtained from static ones which are annotated with upper or lower bars and specify active components of the system at the current time instant. The dynamic expression with upper bar (the overlined one) $\overline{E}$ denotes the *initial*, and that with lower bar (the underlined one) $\underline{E}$ denotes the *final* state of the process specified by a static expression E. The *underlying static expression* of a dynamic one is obtained by removing all upper and lower bars from it.

Definition 2.4. Let $E \in \mathit{StatExpr},\ a \in \mathit{Act}$. A *dynamic expression* of dtsPBC is

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

DynExpr denotes the set of *all dynamic expressions* of dtsPBC.

If the underlying static expression of a dynamic one is not regular, then the corresponding LDTSPN can be non-safe, as Example 4.3 will show (but the LDTSPN is 2-bounded in the worst case (Best *et al.* 2001)).

Definition 2.5. A dynamic expression is *regular* if its underlying static expression is regular.

RegDynExpr denotes the set of *all regular dynamic expressions* of dtsPBC.

3. Operational semantics

In this section, we define the step operational semantics via labeled transition systems. An illustrating example will be given at the end of the section.

3.1. *Inaction rules*

The inaction rules for dynamic expressions describe their structural transformations not changing the states of the specified processes. The goal of these syntactic transformations is to obtain the well-structured terminal expressions called operative ones to which no inaction rules can be further applied. As we shall see, the application of an inaction rule to a dynamic expression does not lead to any discrete time step in the corresponding LDTSPN, hence, no transitions fire and its current marking remains unchanged.

Thus, an application of every inaction rule does not require any time delay, i.e. the dynamic expression transformation described by the rule is accomplished instantly.

In Table 1, we define inaction rules for regular dynamic expressions in the form of overlined and underlined static ones. In this table, $E, F, K \in \mathit{RegStatExpr}$ and $a \in \mathit{Act}$.

In Table 2, we propose inaction rules for regular dynamic expressions in the arbitrary form. In this table, $E, F \in \mathit{RegStatExpr},\ G, H, \widetilde{G}, \widetilde{H} \in \mathit{RegDynExpr}$ and $a \in \mathit{Act}$.

Table 1. *Inaction rules for overlined and underlined regular static expressions*

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

Table 2. *Inaction rules for arbitrary regular dynamic expressions*

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

Definition 3.1. A regular dynamic expression G is *operative* if no inaction rule can be applied to it.

OpRegDynExpr denotes the set of *all operative regular dynamic expressions* of dtsPBC. Any regular dynamic expression can be transformed into a (possibly not unique) operative one by the inaction rules. In the following, we shall consider regular expressions only and omit the word "regular".

Definition 3.2. Let $\approx\, = (\Rightarrow \cup \Leftarrow)^*$ be a structural equivalence of dynamic expressions in dtsPBC. Thus, two dynamic expressions G and G' are *structurally equivalent*, denoted by $G \approx G'$, if they can be reached from each other by applying the inaction rules in a forward or backward direction.

3.2. *Action and empty loop rules*

The action rules are applied when some activities are executed. We also have the empty loop rule which is used to capture a delay of one time unit in the same state when the empty multiset of activities is executed. The action and empty loop rules will be used later to determine all multisets of activities which can be executed from the structural equivalence class of every dynamic expression (i.e. from the state of the corresponding process). This information together with that about probabilities of the activities to be executed from the process state will be used to calculate the probabilities of such executions.

The action rules describe dynamic expression transformations due to execution of non-empty multisets of activities. The rules represent possible state changes of the specified processes when some non-empty multisets of activities are executed. As we shall see, the application of an action rule to a dynamic expression leads to a discrete time step in the corresponding LDTSPN at which some transitions fire and the current marking is changed, unless there is a self-loop produced by the iterative execution of a non-empty multiset (which should be one-element, i.e. the single activity, since we do not allow concurrency at the highest level of the second argument of iteration).

The empty loop rule $G \xrightarrow{\emptyset} G$ describes dynamic expression transformations due to execution of the empty multiset of activities at a discrete time step. The rule reflects a non-zero probability to stay in the current state at the next time moment, which is an essential feature of discrete time stochastic processes. As we shall see, the application of the empty loop rule to a dynamic expression leads to a discrete time step in the corresponding LDTSPN at which no transitions fire and the current marking is not changed. This is a new rule that has no prototype among inaction rules of PBC, since it represents a time delay. The PBC rule $G \xrightarrow{\emptyset} G$ from (Best *et al.* 2001) in our setting would correspond to the rule $G \Rightarrow G$ describing the stay in the current state when no time elapses. Since we do not need the latter rule to transform dynamic expressions into operative ones and it can even destroy the definition of operative expressions, we do not introduce it in dtsPBC.

Thus, an application of every action rule or the empty loop rule requires one discrete time unit delay, i.e. the execution of a (possibly empty) multiset of activities resulting to the dynamic expression transformation described by the rule is accomplished instantly after one unit of time elapses.

In Table 3, we define the action and empty loop rules. In this table, $(\alpha, \rho), (\beta, \chi) \in \mathcal{SL}$, $E, F \in \mathit{RegStatExpr}$, $G, H \in \mathit{OpRegDynExpr}$, $\widetilde{G}, \widetilde{H} \in \mathit{RegDynExpr}$ and $a \in \mathit{Act}$. Moreover, $\Gamma, \Delta \in I\!\!N_f^{\mathcal{SL}} \setminus \{\emptyset\}$ and $\Gamma' \in I\!\!N_f^{\mathcal{SL}}$.

Table 3. *Action and empty loop rules*

El $G \xrightarrow{\emptyset} G$	**B** $\overline{(\alpha, \rho)} \xrightarrow{\{(\alpha,\rho)\}} \underline{(\alpha, \rho)}$	**SC1** $\frac{G \xrightarrow{\Gamma} \widetilde{G},\ \circ \in \{;, []\}}{G \circ E \xrightarrow{\Gamma} \widetilde{G} \circ E}$	**SC2** $\frac{G \xrightarrow{\Gamma} \widetilde{G},\ \circ \in \{;, []\}}{E \circ G \xrightarrow{\Gamma} E \circ \widetilde{G}}$
P1 $\frac{G \xrightarrow{\Gamma} \widetilde{G}}{G \| H \xrightarrow{\Gamma} \widetilde{G} \| H}$	**P2** $\frac{H \xrightarrow{\Gamma} \widetilde{H}}{G \| H \xrightarrow{\Gamma} G \| \widetilde{H}}$	**P3** $\frac{G \xrightarrow{\Gamma} \widetilde{G},\ H \xrightarrow{\Delta} \widetilde{H}}{G \| H \xrightarrow{\Gamma + \Delta} \widetilde{G} \| \widetilde{H}}$	**L** $\frac{G \xrightarrow{\Gamma} \widetilde{G}}{G[f] \xrightarrow{f(\Gamma)} \widetilde{G}[f]}$
Rs $\frac{G \xrightarrow{\Gamma} \widetilde{G},\ a, \hat{a} \not\in \mathcal{A}(\Gamma)}{G \text{ rs } a \xrightarrow{\Gamma} \widetilde{G} \text{ rs } a}$	**I1** $\frac{G \xrightarrow{\Gamma} \widetilde{G}}{[G * E * F] \xrightarrow{\Gamma} [\widetilde{G} * E * F]}$	**I2** $\frac{G \xrightarrow{\Gamma} \widetilde{G}}{[E * G * F] \xrightarrow{\Gamma} [E * \widetilde{G} * F]}$	**I3** $\frac{G \xrightarrow{\Gamma} \widetilde{G}}{[E * F * G] \xrightarrow{\Gamma} [E * F * \widetilde{G}]}$
Sy1 $\frac{G \xrightarrow{\Gamma} \widetilde{G}}{G \text{ sy } a \xrightarrow{\Gamma} \widetilde{G} \text{ sy } a}$	**Sy2** $\frac{G \text{ sy } a \xrightarrow{\Gamma' + \{(\alpha,\rho)\} + \{(\beta,\chi)\}} \widetilde{G} \text{ sy } a,\ a \in \alpha,\ \hat{a} \in \beta}{G \text{ sy } a \xrightarrow{\Gamma' + \{(\alpha \oplus_a \beta, \rho \cdot \chi)\}} \widetilde{G} \text{ sy } a}$		

Almost all the rules in Table 3 (excepting **El**, **P3** and **Sy2**) resemble those of sPBC (Macià *et al.* 2001), but the former correspond to execution of multisets of activities, not of single activities, as in the latter.

Rule **El** corresponds to one discrete time unit delay while executing no activities and therefore it has no analogues among the rules of sPBC that adopts the continuous time model.

Rule **P3** has no similar rules in sPBC, since interleaving semantics of the algebra allows no simultaneous execution of activities. On the other hand, **P3** has in PBC the analogous rule **PAR** that is used to construct step semantics of the calculus, but the former rule corresponds to execution of multisets of activities, unlike that of multisets of multiactions in the latter rule.

Rule **Sy2** differs from the corresponding synchronization rule in sPBC, since the probability of synchronization in the former rule and the rate of synchronization in the latter rule are calculated in a distinct way.

Rule **Sy2** establishes that the synchronization of two stochastic multiactions is made by taking the product of their probabilities, since we are considering that both must occur for the synchronization to happen, so this corresponds, in some sense, to the probability of the independent event intersection, but the real situation is more complex, since these stochastic multiactions can also be executed in parallel. Nevertheless, when scoping (the combined operation consisting of synchronization followed by restriction over the same action (Best *et al.* 2001)) is applied over a parallel execution, we get as final result just the simple product of the probabilities, since no normalization is needed there. Multiplication is an associative and commutative binary operation that is distributive over addition, i.e. it fulfills all practical conditions imposed on the synchronization operator in (Hillston 1994). Further, if both arguments of multiplication are from $(0; 1)$ then the result belongs to the same interval, hence, multiplication naturally maintains probabilistic compositionality in our model. Our approach is similar to the multiplication of rates of the synchronized actions in MTIPP (Hermanns and Rettelbach 1994) in the case when the rates are less than 1. Moreover, for the probabilities ρ and χ of two stochastic multiactions to be synchronized we have $\rho \cdot \chi < \min\{\rho, \chi\}$, i.e. multiplication meets the performance requirement stating that the probability of the resulting synchronized stochastic multiaction should be less than the probabilities of the two ones to be synchronized. While performance evaluation, it is usually supposed that the execution of two components together require more system resources and time than the execution of each single one. This resembles the *bounded capacity* assumption from (Hillston 1994). Thus, multiplication is easy to handle with and it satisfies the algebraic, probabilistic, time and performance requirements. Therefore, we have chosen the product of the probabilities for the synchronization. See also (Brinksma *et al.* 1995; Brinksma and Hermanns 2001) for a discussion about binary operations producing the rates of synchronization in the continuous time setting.

As we shall see, for every LDTSPN obtained by synchronization of two LDTSPNs, this approach allows us to calculate the transition firing probabilities using the standard transition probability function for that net class. If concurrency aspects are not relevant then interleaving semantics is used which abstracts from steps with more than one element. After the abstraction, the probabilities of the remaining one-element steps are normalized to keep the sums of outgoing probabilities equal to one. For two synchronized LDTSPNs, our approach allows us to extract the interleaving probabilities from the step ones in the same way as for two non-synchronized parallel LDTSPNs.

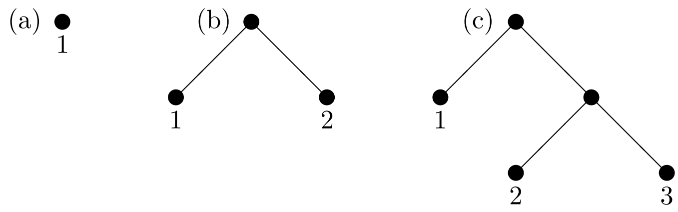

Fig. 1. The binary trees encoded with the numberings 1, (1)(2) and (1)((2)(3))

We do not allow self-synchronization, i.e. synchronization of an activity with itself, to avoid an unexpected behaviour and many technical difficulties (Best *et al.* 2001).

3.3. *Transition systems*

Now we construct labeled probabilistic transition systems associated with dynamic expressions and used to define the operational semantics of dtsPBC.

The expressions of dtsPBC can contain identical activities. To avoid technical difficulties, we must enumerate coinciding activities, for instance, from left to right in the syntax of expressions. The new activities resulted from synchronization will be annotated with concatenation of numberings of the activities they come from, hence, the numbering should have a tree structure to reflect the effect of multiple synchronizations. We define the numbering which encodes a binary tree with the leaves labeled by natural numbers.

Definition 3.3. The *numbering* of expressions is defined as $\iota ::= \ n \mid (\iota)(\iota)$, where $n \in \mathbb{N}$.

Num denotes the set of *all numberings* of expressions.

Example 3.1. The numbering 1 encodes the binary tree depicted in Figure 1(a) with the root labeled by 1. The numbering (1)(2) corresponds to the binary tree depicted in Figure 1(b) without internal nodes and with two leaves labeled by 1 and 2. The numbering (1)((2)(3)) represents the binary tree depicted in Figure 1(c) with one internal node, which is the root for the subtree (2)(3), and three leaves labeled by $1, 2$ and 3.

The new activities resulting from applications of the second rule for synchronization **Sy2** in different orders should be considered up to permutation of their numbering. In this way, we shall recognize different instances of the same activity. If we compare the contents of different numberings, i.e. the sets of natural numbers in them, we shall be able to identify the mentioned instances. The *content* of a numbering $\iota \in Num$ is defined as $Cont(\iota) = \begin{cases} \{\iota\}, & \text{if } \iota \in \mathbb{N}; \\ Cont(\iota_1) \cup Cont(\iota_2), & \text{if } \iota = (\iota_1)(\iota_2). \end{cases}$
After we apply the enumeration, the multisets of activities from the expressions become the proper sets. In the following, we suppose that the identical activities are enumerated when needed to avoid ambiguity. This enumeration is considered to be implicit.

Definition 3.4. Let G be a dynamic expression. Then $[G]_{\approx} = \{H \mid G \approx H\}$ is the equivalence class of G w.r.t. the structural equivalence. The *derivation set* of a dynamic expression G, denoted by $DR(G)$, is the minimal set such that

— $[G]_\approx \in DR(G)$;

— if $[H]_\approx \in DR(G)$ and there exists Γ such that $H \xrightarrow{\Gamma} \widetilde{H}$ then $[\widetilde{H}]_\approx \in DR(G)$.

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

The set of *all multisets of activities executable in* s is $Exec(s) = \{\Gamma \mid \exists H \in s\ \exists \widetilde{H}\ H \xrightarrow{\Gamma} \widetilde{H}\}$. Note that if $\Gamma \in Exec(s)$ then by rules **P3**, **Sy2** and definition of $Exec(s)$ for all $\Delta \subseteq \Gamma$ we have $\Delta \in Exec(s)$.

Let $\Gamma \in Exec(s) \setminus \{\emptyset\}$. The *probability that the multiset of activities* Γ *is ready for execution in* s is

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

In the case $\Gamma = \emptyset$ we define

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

Thus, if $Exec(s) \neq \{\emptyset\}$ then $PF(\Gamma, s)$ can be interpreted as a *joint* probability of independent events. Each such an event consists in the positive or negative decision to be executed of a particular activity. Every executable activity decides probabilistically (using its probabilistic part) and independently (from others), if it wants to be executed in s. If Γ is a multiset of all executable activities which have decided to be executed in s and $\Gamma \in Exec(s)$ then Γ is ready for execution in s. The multiplication in the definition is used because it reflects the probability of the independent event intersection. Alternatively, when $\Gamma \neq \emptyset$, $PF(\Gamma, s)$ can be interpreted as the probability to execute *exclusively* the multiset of activities Γ in s, i.e. the probability of *intersection* of two events calculated using the conditional probability formula in the form $\mathsf{P}(X \cap Y) = \mathsf{P}(X|Y)\mathsf{P}(Y)$. The event X consists in the execution of Γ in s. The event Y consists in the non-execution in s of all the executable activities not belonging to Γ. Since the mentioned non-executions are obviously independent events, the probability of Y is a product of the probabilities of the non-executions: $\mathsf{P}(Y) = \prod_{\{\{(\beta,\chi)\}\in Exec(s)\mid(\beta,\chi)\not\in\Gamma\}}(1-\chi)$. The conditioning of X by Y makes the executions of the activities from Γ independent, since all of them can be executed in parallel in s by definition of $Exec(s)$. Hence, the probability to execute Γ *under condition* that no executable activities not belonging to Γ are executed in s is a product of probabilities of these activities: $\mathsf{P}(X|Y) = \prod_{(\alpha,\rho)\in\Gamma}\rho$. Thus, the probability that Γ is executed *and* no executable activities not belonging to Γ are executed in s is the probability of X conditioned by Y multiplied by the probability of Y: $\mathsf{P}(X \cap Y) = \mathsf{P}(X|Y)\mathsf{P}(Y) = \prod_{(\alpha,\rho)\in\Gamma}\rho \cdot \prod_{\{\{(\beta,\chi)\}\in Exec(s)\mid(\beta,\chi)\not\in\Gamma\}}(1-\chi)$. When $\Gamma = \emptyset$, $PF(\Gamma, s)$ can be interpreted as the probability not to execute in s any executable activities, thus, $PF(\emptyset, s) = \prod_{\{(\beta,\chi)\}\in Exec(s)}(1-\chi)$. When only the empty multiset of activities can be executed in s, i.e. $Exec(s) = \{\emptyset\}$, we have $PF(\emptyset, s) = 1$, since we stay in s in this case.

Let $\Gamma \in Exec(s)$. Besides Γ, some other multisets of activities may be ready for execution in s, hence, a kind of conditioning or normalization is needed to calculate the execution probability. The *probability to execute the multiset of activities* Γ *in* s is

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

Thus, $PT(\Gamma, s)$ can be interpreted as the *conditional* probability to execute Γ in s calculated using the conditional probability formula in the form $\mathsf{P}(Z|W) = \frac{\mathsf{P}(Z \cap W)}{\mathsf{P}(W)}$. The event Z consists in the exclusive execution of Γ in s, hence, $\mathsf{P}(Z) = PF(\Gamma, s)$. The event W consists in the exclusive execution of any multiset (including the empty one) $\Delta \in Exec(s)$ in s. Thus, $W = \cup_j Z_j$, where for all j, Z_j are mutually exclusive events and there exists i such that $Z = Z_i$. We have $\mathsf{P}(W) = \sum_j \mathsf{P}(Z_j) = \sum_{\Delta \in Exec(s)} PF(\Delta, s)$, because summation reflects the probability of the mutually exclusive event union. Since $Z \cap W = Z_i \cap (\cup_j Z_j) = Z_i = Z$, we have $\mathsf{P}(Z|W) = \frac{\mathsf{P}(Z)}{\mathsf{P}(W)} = \frac{PF(\Gamma, s)}{\sum_{\Delta \in Exec(s)} PF(\Delta, s)}$. $PF(\Gamma, s)$ can also be seen as the *potential* probability to execute Γ in s, since we have $PF(\Gamma, s) = PT(\Gamma, s)$ only when *all* multisets (including the empty one) consisting of the executable activities can be executed in s. In this case, all the mentioned activities can be executed in parallel in s and we have $\sum_{\Delta \in Exec(s)} PF(\Delta, s) = 1$, since this sum collects the products of *all* combinations of the probability parts of the activities and the negations of these parts. But in general, for example, for two activities (α, ρ) and (β, χ) executable in s, it may happen that they cannot be executed in s together, in parallel, i.e. $\emptyset, \{(\alpha, \rho)\}, \{(\beta, \chi)\} \in Exec(s)$, but $\{(\alpha, \rho), (\beta, \chi)\} \notin Exec(s)$. Note that $PT(\emptyset, s) \in (0; 1]$, hence, there is a non-zero probability to stay in the state s at the next time moment, and the residence time in s is at least 1 discrete time unit.

Note that the sum of outgoing probabilities for the expressions belonging to the derivations of G is equal to 1, i.e. for all $s \in DR(G)$ we have $\sum_{\Gamma \in Exec(s)} PT(\Gamma, s) = 1$. This follows from the definition of $PT(\Gamma, s)$ and guarantees that $PT(\Gamma, s)$ defines a probability distribution.

The *probability to move from* s *to* $\tilde{s}$ *by executing any multiset of activities* is

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

Example 3.2. Let $E = (\{a\}, \rho)[](\{a\}, \chi)$. $DR(\overline{E})$ consists of the equivalence classes $s_1 = [\overline{E}]_\approx$ and $s_2 = [\underline{E}]_\approx$. The execution probabilities are calculated as follows. Since $Exec(s_1) = \{\emptyset, \{(\{a\}, \rho)\}, \{(\{a\}, \chi)\}\}$, we get $PF(\{(\{a\}, \rho)\}, s_1) = \rho(1 - \chi)$, $PF(\{(\{a\}, \chi)\}, s_1) = \chi(1 - \rho)$ and $PF(\emptyset, s_1) = (1 - \rho)(1 - \chi)$.
Then $\sum_{\Delta \in Exec(s_1)} PF(\Delta, s_1) = \rho(1 - \chi) + \chi(1 - \rho) + (1 - \rho)(1 - \chi) = 1 - \rho\chi$. Thus, $PT(\{(\{a\}, \rho)\}, s_1) = \frac{\rho(1-\chi)}{1-\rho\chi}$, $PT(\{(\{a\}, \chi)\}, s_1) = \frac{\chi(1-\rho)}{1-\rho\chi}$ and $PT(\emptyset, s_1) = PM(s_1, s_1) = \frac{(1-\rho)(1-\chi)}{1-\rho\chi}$. Further, $Exec(s_2) = \{\emptyset\}$, hence, $\sum_{\Delta \in Exec(s_2)} PF(\Delta, s_2) = PF(\emptyset, s_2) = 1$ and $PT(\emptyset, s_2) = PM(s_2, s_2) = \frac{1}{1} = 1$. Finally, $PM(s_1, s_2) = PT(\{(\{a\}, \rho)\}, s_1) + PT(\{(\{a\}, \chi)\}, s_1) = \frac{\rho(1-\chi)}{1-\rho\chi} + \frac{\chi(1-\rho)}{1-\rho\chi} = \frac{\rho+\chi-2\rho\chi}{1-\rho\chi}$.

Definition 3.5. 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 $L_G = I\!N_f^{\mathcal{SL}} \times (0; 1]$;
— the set of *transitions* is $\mathcal{T}_G = \{(s, (\Gamma, PT(\Gamma, s)), \tilde{s}) \mid s, \tilde{s} \in DR(G),\ \exists H \in s\ \exists \widetilde{H} \in \tilde{s}\ H \xrightarrow{\Gamma} \widetilde{H}\}$;
— the *initial state* is $s_G = [G]_\approx$.

The definition of $TS(G)$ is correct, i.e. for every state, the sum of the probabilities of all the transitions starting from it is 1. This is guaranteed by the note after the definition of $PT(\Gamma, s)$. Thus, we have defined a *generative* model of probabilistic processes (Jou and Smolka 1990), according to the classification from (van Glabbeek *et al.* 1995). The reason is that the sum of the probabilities of the transitions with all possible labels should be equal to 1, not only of those with the same labels (up to enumeration of activities they include) as in the *reactive* models (Larsen and Skou 1991), and we do not have the nested probabilistic choice as in the *stratified* models (van Glabbeek *et al.* 1995).

The transition system $TS(G)$ of a dynamic expression G describes all steps that occur at discrete time moments with some (one-step) probability and that consist of multisets of activities. Every step occurs instantly after one discrete time unit delay, the step can change the current state to another one. The states are the structural equivalence classes of dynamic expressions obtained by application of action rules starting from the expressions belonging to $[G]_\approx$. A transition $(s, (\Gamma, \mathcal{P}), \tilde{s}) \in \mathcal{T}_G$ is written as $s \xrightarrow{\Gamma}_{\mathcal{P}} \tilde{s}$ and interpreted as follows: the probability to change the state s to $\tilde{s}$ by executing Γ is $\mathcal{P}$.

Note that Γ can be the empty multiset, and its execution does not change the current state (the equivalence class), since we have a loop transition $s \xrightarrow{\emptyset}_{\mathcal{P}} s$ from a state s to itself as a result of executing the empty multiset. This corresponds to application of the empty loop rule to expressions from the equivalence class. 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 $(0; 1)$. The step probabilities belong to the interval $(0; 1]$. The step probability is 1 when we cannot leave a state s, hence, there exists only one transition from it, namely, the empty loop transition $s \xrightarrow{\emptyset}_1 s$.

We write $s \xrightarrow{\Gamma} \tilde{s}$ if there exists $\mathcal{P}$ such that $s \xrightarrow{\Gamma}_{\mathcal{P}} \tilde{s}$ and $s \to \tilde{s}$ if there exists Γ such that $s \xrightarrow{\Gamma} \tilde{s}$. For a one-element multiset of activities $\Gamma = \{(\alpha, \rho)\}$, we write $s \xrightarrow{(\alpha,\rho)}_{\mathcal{P}} \tilde{s}$ and $s \xrightarrow{(\alpha,\rho)} \tilde{s}$.

Isomorphism is a coincidence of systems up to renaming of their components. Let $\simeq$ denote isomorphism between transition systems that binds their initial states.

Definition 3.6. Dynamic expressions G and G' are *equivalent w.r.t. transition systems*, denoted by $G =_{ts} G'$, if $TS(G) \simeq TS(G')$.

Definition 3.7. Let G be a dynamic expression. The *underlying discrete time Markov chain (DTMC)* of G, denoted by $DTMC(G)$, has the state space $DR(G)$, the initial state $[G]_\approx$ and the transitions $s \to_{\mathcal{P}} \tilde{s}$, if $s \to \tilde{s}$ and $\mathcal{P} = PM(s, \tilde{s})$.

For a dynamic expression G, a discrete random variable is associated with every state of $DTMC(G)$. The variable captures a residence time in the state. One can interpret staying in a state at the next discrete time moment as a failure and leaving it as a success of some trial series. It is easy to see that the random variables are geometrically distributed with the parameter $1-PM(s,s)$, since the probability to stay in the state $s \in DR(G)$ for $k-1$ time moments and leave it at the moment $k \geq 1$ is $PM(s,s)^{k-1}(1-PM(s,s))$ (the residence time is k in this case). The mean value formula for the geometrical distribution allows us to calculate the *average sojourn time in the state* s as $SJ(s) = \frac{1}{1-PM(s,s)}$. The *average sojourn time vector* of G, denoted by SJ, has the elements $SJ(s),\ s \in DR(G)$. Analogously, the *sojourn time variance in the state* s is $VAR(s) = \frac{PM(s,s)}{(1-PM(s,s))^2}$. The *sojourn time variance vector* of G, denoted by VAR, has the elements $VAR(s),\ s \in DR(G)$.

Example 3.3. 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 follows: $E = [((\{a\},\rho)_1[](\{a\},\rho)_2) * (\{b\},\chi) * (\{c\},\theta)]$. In Figure 2, the transition system $TS(\overline{E})$ and the underlying DTMC $DTMC(\overline{E})$ are presented. For simplicity, the states are labeled by expressions belonging to the corresponding equivalence classes, and singleton multisets of activities are written without braces.

$DR(\overline{E})$ consists of the equivalence classes $s_1 = [\overline{[E_1 * E_2 * E_3]}]_{\approx},\ s_2 = [[E_1 * \overline{E_2} * E_3]]_{\approx},\ s_3 = [\underline{[E_1 * E_2 * E_3]}]_{\approx}$. Let us demonstrate how the transition probabilities are calculated. For instance, we have $PF(\{(\{a\},\rho)_1\},s_1) = PF(\{(\{a\},\rho)_2\},s_1) = \rho(1-\rho)$ and $PF(\emptyset,s_1) = (1-\rho)^2$. Hence, $\sum_{\Delta \in Exec(s_1)} PF(\Delta,s_1) = 2\rho(1-\rho)+(1-\rho)^2 = 1-\rho^2$. Thus, $PT(\{(\{a\},\rho)_1\},s_1) = PT(\{(\{a\},\rho)_2\},s_1) = \frac{\rho(1-\rho)}{1-\rho^2} = \frac{\rho(1-\rho)}{(1-\rho)(1+\rho)} = \frac{\rho}{1+\rho}$ and $PT(\emptyset,s_1) = \frac{(1-\rho)^2}{1-\rho^2} = \frac{(1-\rho)^2}{(1-\rho)(1+\rho)} = \frac{1-\rho}{1+\rho}$. Other probabilities are calculated similarly. The average sojourn time vector of $\overline{E}$ is $SJ = \left(\frac{1+\rho}{2\rho}, \frac{1-\chi\theta}{\theta(1-\chi)}, \infty\right)$. The sojourn time variance vector of $\overline{E}$ is $VAR = \left(\frac{1-\rho^2}{4\rho^2}, \frac{(1-\theta)(1-\chi\theta)}{\theta^2(1-\chi)^2}, \infty\right)$.

4. Denotational semantics

In this section, we define the denotational semantics in terms of a subclass of LDTSPNs, called discrete time stochastic Petri boxes (dts-boxes). An illustrating example will be given at the end of the section.

4.1. *Labeled DTSPNs*

We introduce a class of labeled discrete time stochastic PNs (LDTSPNs), a subclass of DTSPNs (Molloy 1985) (we do not allow the transition probabilities to be equal to 1) with transition labeling. LDTSPNs are somewhat similar to labeled weighted DTSPNs (LWDTSPNs) from (Buchholz and Tarasyuk 2001), but in LWDTSPNs all transitions have weights, the transition probabilities may be equal to 1 and only maximal fireable subsets of the enabled transitions are fired.

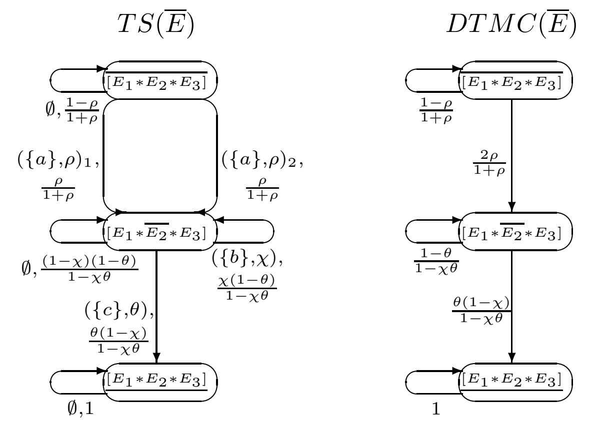

Fig. 2. The transition system and the underlying DTMC of $\overline{E}$ for $E = [((\{a\},\rho)_1[]$ $(\{a\},\rho)_2) * (\{b\},\chi) * (\{c\},\theta)]$

Definition 4.1. A *labeled DTSPN (LDTSPN)* is a tuple $N = (P_N, T_N, W_N, \Omega_N, L_N, M_N)$, where

— P_N and T_N are finite sets of *places* and *transitions*, respectively, such that $P_N \cup T_N \neq \emptyset$ and $P_N \cap T_N = \emptyset$;
— $W_N : (P_N \times T_N) \cup (T_N \times P_N) \to I\!N$ is a function providing the *weights of arcs* between places and transitions;
— $\Omega_N : T_N \to (0;1)$ is the *transition probability* function associating transitions with probabilities;
— $L_N : T_N \to \mathcal{L}$ is the *transition labeling* function assigning multiactions to transitions;
— $M_N \in I\!N_f^{P_N}$ is the *initial marking.*

A graphical representation of LDTSPNs is like that for standard labeled PNs, but with probabilities written near the corresponding transitions. In the case the probabilities are not given in the picture, they are considered to be of no importance. The weights of arcs are depicted near them. The names of places and transitions are depicted near them when needed.

Let N be an LDTSPN and $t \in T_N$, $U \in I\!N_f^{T_N}$. 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 $U^\bullet = \sum_{t \in U} t^\bullet$. Note that for $U = \emptyset$ we have ${}^\bullet\emptyset = \emptyset = \emptyset^\bullet$.

A transition $t \in T_N$ is *enabled* in a marking $M \in I\!N_f^{P_N}$ of LDTSPN N if ${}^\bullet t \subseteq M$. Let $Ena(M)$ be the set of *all transitions (such that each of them is) enabled in a marking* M. A set of transitions $U \subseteq Ena(M)$ is *enabled* in a marking M if ${}^\bullet U \subseteq M$. Firings of transitions are atomic operations, and transitions may fire concurrently in steps. We assume that all transitions participating in a step should differ, hence, only the sets (not multisets) of transitions may fire. Thus, we do not allow self-concurrency, i.e. firing of transitions concurrently to themselves. This restriction is introduced because we would like to avoid technical difficulties while calculating probabilities for multisets of transitions as we shall see after the following formal definitions. Moreover, we do not need to consider self-concurrency, since denotational semantics of expressions will be defined via dts-boxes which are safe LDTSPNs (hence, no self-concurrency is possible).

Let M be a marking of an LDTSPN N. A transition $t \in Ena(M)$ fires with probability $\Omega_N(t)$ when no other transitions conflicting with it are enabled.

Let $U \subseteq Ena(M),\ U \neq \emptyset$ and ${}^\bullet U \subseteq M$. The *probability that the set of transitions U is ready for firing in M* is

$$PF(U, M) = \prod_{t \in U} \Omega_N(t) \cdot \prod_{u \in Ena(M) \setminus U} (1 - \Omega_N(u)).$$

In the case $U = \emptyset$ we define

$$PF(\emptyset, M) = \begin{cases} \prod_{u \in Ena(M)} (1 - \Omega_N(u)), & \text{if } Ena(M) \neq \emptyset; \\ 1, & \text{otherwise.} \end{cases}$$

Let $U \subseteq Ena(M)$ and ${}^\bullet U \subseteq M$. Besides U, some other sets of transitions may be ready for firing in M, hence, a kind of conditioning or normalization is needed to calculate the firing probability. The concurrent firing of the transitions from U changes the marking M to $\widetilde{M} = M - {}^\bullet U + U^\bullet$, denoted by $M \stackrel{U}{\rightarrow}_{\mathcal{P}} \widetilde{M}$, where $\mathcal{P} = PT(U, M)$ is the *probability that the set of transitions U fires in M* defined as

$$PT(U, M) = \frac{PF(U, M)}{\sum_{\{V \mid {}^\bullet V \subseteq M\}} PF(V, M)}.$$

Note that in the case $U = \emptyset$ we have $M = \widetilde{M}$.

Let $Ena(M) = \{t_1, \ldots, t_n\}$ be a mutually exclusive set of transitions (i.e. firing of any transition from the set results in a marking in which no other transition from the set is enabled) and $\rho_i = \Omega_N(t_i)$ $(1 \leq i \leq n)$. Then $PT(\{t_i\}, M)$ resembles the probabilistic function $P[E_i]$ from (Molloy 1985), which defines the probability of the event E_i, that transition t_i in a mutually exclusive set of transitions $\{t_1, \ldots, t_n\}$ will fire in the marking M. We have $P[E_i] = \frac{\frac{\rho_i}{1-\rho_i}}{1+\sum_{j=1}^{n}\frac{\rho_j}{1-\rho_j}} = \frac{\frac{\rho_i(1-\rho_1)\cdots(1-\rho_n)}{1-\rho_i}}{(1-\rho_1)\cdots(1-\rho_n)+\sum_{j=1}^{n}\frac{\rho_j(1-\rho_1)\cdots(1-\rho_n)}{1-\rho_j}}$, where $\frac{\rho_i(1-\rho_1)\cdots(1-\rho_n)}{1-\rho_i}$ corresponds to $PF(\{t_i\}, M)$ in our setting. Further, $PT(\emptyset, M)$ resembles the probabilistic function $P[E_0]$, which defines the probability of the event E_0, that no transitions from the mutually exclusive set of transitions $\{t_1, \ldots, t_n\}$ will fire in the marking M. We have $P[E_0] = \frac{1}{1+\sum_{j=1}^{n}\frac{\rho_j}{1-\rho_j}} = \frac{(1-\rho_1)\cdots(1-\rho_n)}{(1-\rho_1)\cdots(1-\rho_n)+\sum_{j=1}^{n}\frac{\rho_j(1-\rho_1)\cdots(1-\rho_n)}{1-\rho_j}}$, where $(1-\rho_1)\cdots(1-\rho_n)$ corresponds to $PF(\emptyset, M)$ in our setting. If $Ena(M)$ is not a mutually exclusive set of transitions, our way to define $PT(U, M)$ for $U \subseteq Ena(M),\ U \neq \emptyset$, also extends the approach of (Molloy 1985; Molloy 1981). The advantage of our two-stage definition of $PT(U, M)$ is that it has a closed form and we do not need to consider which sets of transitions are exclusive, instead, we just consider the probability that U fires in M under condition that only particular subsets of $Ena(M)$ can fire in M.

Note that for all markings of an LDTSPN N the sum of outgoing probabilities is equal to 1, i.e. for all $M \in \mathbb{N}_f^{P_N}$ we have $PT(\emptyset, M) + \sum_{\{U \mid {}^\bullet U \subseteq M\}} PT(U, M) = 1$. This follows from the definition of $PT(U, M)$ and guarantees that it defines a probability distribution.

We write $M \xrightarrow{U} \widetilde{M}$ if there exists $\mathcal{P}$ such that $M \xrightarrow{U}_{\mathcal{P}} \widetilde{M}$ and $M \to \widetilde{M}$ if there exists U such that $M \xrightarrow{U} \widetilde{M}$.

Definition 4.2. Let N be an LDTSPN.

— The *reachability set* $RS(N)$ of N is the minimal set of markings such that
  - $M_N \in RS(N)$;
  - if $M \in RS(N)$ and $M \to \widetilde{M}$ then $\widetilde{M} \in RS(N)$.

— The *reachability graph* $RG(N)$ of N is a directed labeled graph with the set of nodes $RS(N)$ and an arc labeled with $(U, \mathcal{P})$ from node M to $\widetilde{M}$ if $M \xrightarrow{U}_{\mathcal{P}} \widetilde{M}$.

— The *underlying discrete time Markov chain (DTMC)* $DTMC(N)$ of N has the state space $RS(N)$, the initial state M_N and the transitions $M \to_{\mathcal{P}} \widetilde{M}$, if $M \to \widetilde{M}$, where $\mathcal{P} = PM(M, \widetilde{M})$ is the *probability to move from* M *to* $\widetilde{M}$ *by firing any set of transitions* defined as

$$PM(M, \widetilde{M}) = \sum_{\{U \mid M \xrightarrow{U} \widetilde{M}\}} PT(U, M).$$

Let N be an LDTSPN and $M \in RS(N)$. The *average sojourn time in the marking* M is $SJ(M) = \frac{1}{1 - PM(M,M)}$. The *average sojourn time vector* of N, denoted by SJ, has the elements $SJ(M),\ M \in RS(N)$. The *sojourn time variance in the marking* M is $VAR(M) = \frac{PM(M,M)}{(1 - PM(M,M))^2}$. The *sojourn time variance vector* of N, denoted by VAR, has the elements $VAR(M),\ M \in RS(N)$.

Example 4.1. In Figure 3, an LDTSPN N with two visible transitions t_1 (labeled by $\{a\}$), t_2 (labeled by $\{b\}$) and one invisible transition t_3 (labeled by $\emptyset$) is presented. Transition probabilities of N are denoted by $\rho = \Omega_N(t_1),\ \chi = \Omega_N(t_2),\ \theta = \Omega_N(t_3)$. In the figure one can see the reachability graph $RG(N)$ and the underlying DTMC $DTMC(N)$ as well. $RS(N)$ consists of the markings $M_1 = (1, 1, 0),\ M_2 = (0, 1, 1),\ M_3 = (1, 0, 1),\ M_4 = (0, 0, 2)$. The average sojourn time vector of N is $SJ = \left(\frac{1}{\rho + \chi - \rho\chi}, \frac{1}{\chi}, \frac{1}{\rho}, \frac{1}{\theta}\right)$. The sojourn time variance vector of N is $VAR = \left(\frac{1 - \rho - \chi + \rho\chi}{(\rho + \chi - \rho\chi)^2}, \frac{1 - \chi}{\chi^2}, \frac{1 - \rho}{\rho^2}, \frac{1 - \theta}{\theta^2}\right)$.

4.2. *Algebra of dts-boxes*

Now we propose discrete time stochastic Petri boxes and associated algebraic operations to define a net representation of dtsPBC expressions.

Definition 4.3. A *discrete time stochastic Petri box (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, such that $P_N \cup T_N \neq \emptyset$ and $P_N \cap T_N = \emptyset$;

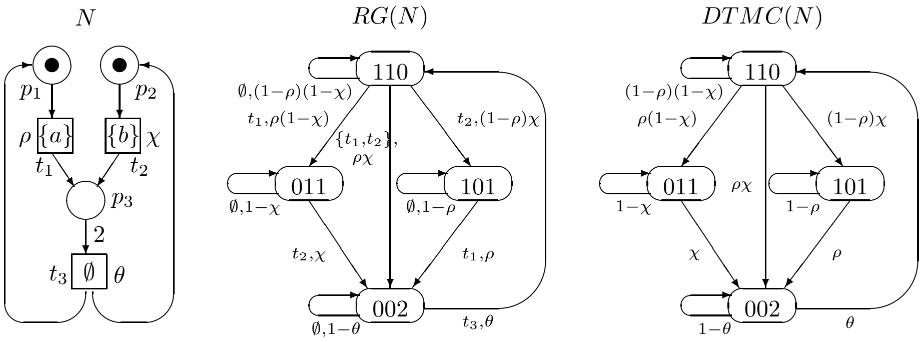

Fig. 3. LDTSPN, its reachability graph and the underlying DTMC

— $W_N : (P_N \times T_N) \cup (T_N \times P_N) \to I\!N$ is a function providing the *weights of arcs* between places and transitions;

— Λ_N is the *place and transition labeling* function such that

- $\Lambda_N|_{P_N} : P_N \to \{\mathsf{e}, \mathsf{i}, \mathsf{x}\}$ (it specifies *entry, internal* and *exit* places, respectively);
- $\Lambda_N|_{T_N} : T_N \to \{\varrho \mid \varrho \subseteq I\!N_f^{\mathcal{SL}} \times \mathcal{SL}\}$ (it associates transitions with the *relabeling relations* on activities).

We require that for all $t \in T_N$ we have ${}^\bullet t \neq \emptyset \neq t^\bullet$. 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 $N^\circ = \{p \in P_N \mid \Lambda_N(p) = \mathsf{x}\}$, it holds: ${}^\circ N \neq \emptyset \neq N^\circ$, ${}^\bullet({}^\circ N) = \emptyset = (N^\circ)^\bullet$.

A dts-box is *plain* if for all $t \in T_N$ we have $\Lambda_N(t) \in \mathcal{SL}$, i.e. $\Lambda_N(t)$ is a constant relabeling that will be defined later. In case of the constant relabeling, the shorthand notation (by an activity) for $\Lambda_N(t)$ will be used. 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 its marking. We use the following notation: $\overline{N} = (N, {}^\circ N)$ and $\underline{N} = (N, N^\circ)$. A marked plain dts-box $(P_N, T_N, W_N, \Lambda_N, M_N)$ could be interpreted as the LDTSPN $(P_N, T_N, W_N, \Omega_N, L_N, M_N)$, where functions Ω_N and L_N are defined as follows: for all $t \in T_N$ we have $\Omega_N(t) = \Omega(\Lambda_N(t))$ and $L_N(t) = \mathcal{L}(\Lambda_N(t))$. Behaviour of the marked dts-boxes follows from the firing rule of LDTSPNs. A plain dts-box N is *n-bounded* ($n \in I\!N$) if $\overline{N}$ is so, i.e. for all $M \in RS(\overline{N})$ and for all $p \in P_N$ we have $M(p) \leq n$, and it is *safe* if it is 1-bounded. A plain dts-box N is *clean* if for all $M \in RS(\overline{N})$ we have ${}^\circ N \subseteq M$ implies $M = {}^\circ N$ and $N^\circ \subseteq M$ implies $M = N^\circ$ i.e. if there are tokens in all its entry (exit) places then no other places have tokens.

The structure of the plain dts-box corresponding to a static expression is constructed like in PBC (Best and Koutny 1995; Best *et al.* 2001), i.e. we use a 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. Thus, the resulting plain dts-boxes are safe and clean. In the definition of the denotational semantics, we shall apply standard constructions used for PBC. Let Θ denote an *operator box* and u denote a *transition name* from PBC setting.

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

— $\varrho_{id} = \{(\{(\alpha, \rho)\}, (\alpha, \rho)) \mid (\alpha, \rho) \in \mathcal{SL}\}$ is the *identity relabeling* keeping the interface;

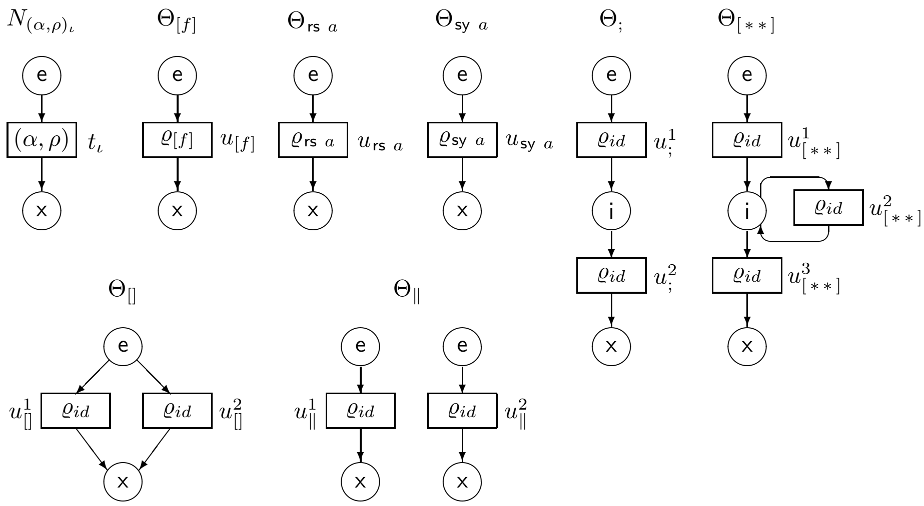

Fig. 4. The plain and operator dts-boxes

— $\varrho_{(\alpha,\rho)} = \{(\emptyset, (\alpha,\rho))\}$ is the *constant relabeling* identified with $(\alpha,\rho) \in \mathcal{SL}$ itself;
— $\varrho_{[f]} = \{(\{(\alpha,\rho)\}, (f(\alpha),\rho)) \mid (\alpha,\rho) \in \mathcal{SL}\}$;
— $\varrho_{\mathsf{rs}\ a} = \{(\{(\alpha,\rho)\}, (\alpha,\rho)) \mid (\alpha,\rho) \in \mathcal{SL},\ a, \hat{a} \notin \alpha\}$;
— $\varrho_{\mathsf{sy}\ a}$ is the least relabeling relation containing ϱ_{id} such that if $(\Gamma, (\alpha,\rho)), (\Delta, (\beta,\chi)) \in \varrho_{\mathsf{sy}\ a}$ and $a \in \alpha,\ \hat{a} \in \beta$, then $(\Gamma + \Delta, (\alpha \oplus_a \beta, \rho \cdot \chi)) \in \varrho_{\mathsf{sy}\ a}$.

The plain and operator dts-boxes are presented in Figure 4. The symbol i is often omitted.

To construct a semantic function associating a plain dts-box with every static expression of dtsPBC, we need to propose the *enumeration* function $Enu : T_N \to Num$. It associates numberings with transitions of the plain dts-box N according to those of activities. In the case of synchronization, the function associates concatenation of the parenthesized numberings of the synchronized transitions with a resulting new transition.

Now we define the enumeration function Enu for every operator of dtsPBC. Let $Box_{dts}(E) = (P_E, T_E, W_E, \Lambda_E)$ be the plain dts-box corresponding to a static expression E, and $Enu_E : T_E \to Num$ be the enumeration function for $Box_{dts}(E)$. We shall use the analogous notation for static expressions F and K.

— $Box_{dts}(E \circ F) = \Theta_\circ(Box_{dts}(E), Box_{dts}(F)),\ \circ \in \{;, [], \|\}$. Since we do not introduce any new transitions, we preserve the initial numbering:

$$Enu(t) = \begin{cases} Enu_E(t), & \text{if } t \in T_E; \\ Enu_F(t), & \text{if } t \in T_F. \end{cases}$$

— $Box_{dts}(E[f]) = \Theta_{[f]}(Box_{dts}(E))$. Since we only replace the labels of some multiactions by a bijection, we preserve the initial numbering:

$$Enu(t) = Enu_E(t),\ t \in T_E.$$

— $Box_{dts}(E \text{ rs } a) = \Theta_{\text{rs } a}(Box_{dts}(E))$. Since we remove all transitions labeled with multiactions containing a or $\hat{a}$, this does not change the numbering of the remaining transitions:

$$Enu(t) = Enu_E(t),\ t \in T_E,\ a, \hat{a} \notin \mathcal{L}(\Lambda_E(t)).$$

— $Box_{dts}(E \text{ sy } a) = \Theta_{\text{sy } a}(Box_{dts}(E))$. Note that for all $v, w \in T_E$ such that $\Lambda_E(v) = (\alpha, \rho),\ \Lambda_E(w) = (\beta, \chi)$ and $a \in \alpha,\ \hat{a} \in \beta$, the new transition t resulting from synchronization of v and w has the label $\Lambda(t) = (\alpha \oplus_a \beta, \rho \cdot \chi)$ and the numbering $Enu(t) = (Enu_E(v))(Enu_E(w))$. The enumeration function is

$$Enu(t) = \begin{cases} Enu_E(t), & \text{if } t \in T_E; \\ (Enu_E(v))(Enu_E(w)), & \text{if } t \text{ results from synchronization of } v \text{ and } w. \end{cases}$$

When we synchronize the same set of transitions in different orders, we get several resulting transitions with the same label and probability, but with different numberings having the same content. Then we shall consider only a single transition from the resulting ones in the plain dts-box to avoid introducing redundant ones.

— $Box_{dts}([E * F * K]) = \Theta_{[**]}(Box_{dts}(E), Box_{dts}(F), Box_{dts}(K))$. Since we do not introduce any new transitions, we preserve the initial numbering:

$$Enu(t) = \begin{cases} Enu_E(t), & \text{if } t \in T_E; \\ Enu_F(t), & \text{if } t \in T_F; \\ Enu_K(t), & \text{if } t \in T_K. \end{cases}$$

Definition 4.4. 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 $Box_{dts}((\alpha, \rho)_\iota) = N_{(\alpha,\rho)_\iota}$;
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 $Box_{dts}(E \circ a) = \Theta_{\circ a}(Box_{dts}(E)),\ \circ \in \{\text{rs}, \text{sy}\}$;
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 $Box_{dts}(\underline{E}) = \underline{Box_{dts}(E)}$. This definition is compositional in the sense that, for any dynamic expression, we may decompose it in some inner dynamic and static expressions, for which we may apply the definition, thus obtaining the corresponding plain dts-boxes, which can be joined according to the term structure (by definition of Box_{dts}), the resulting plain box being marked in the places that were marked in the argument nets.

Let $\simeq$ denote the isomorphism between transition systems and reachability graphs or between DTMCs that binds their initial states. 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 $DTMC(N)$.

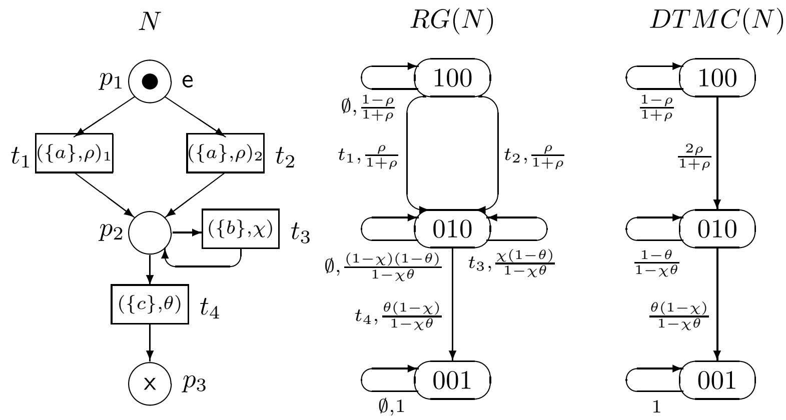

Fig. 5. The marked dts-box $N = Box_{dts}(\overline{E})$ for $E = [((\{a\},\rho)_1[](\{a\},\rho)_2) * (\{b\},\chi) * (\{c\},\theta)]$, its reachability graph and the underlying DTMC

Theorem 4.1. (Tarasyuk 2006) For any static expression E

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

Proof. As 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 analogous probability functions. □

Proposition 4.1. (Tarasyuk 2006) For any static expression E

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

Proof. By Theorem 4.1 and definitions of underlying DTMCs for dynamic expressions and LDTSPNs, since transition probabilities of the associated DTMCs are the sums of those belonging to transition systems or reachability graphs.

□

Example 4.2. Let E be from Example 3.3. In Figure 5, the marked dts-box $N = Box_{dts}(\overline{E})$, its reachability graph $RG(N)$ and the underlying DTMC $DTMC(N)$ are presented. It is easy to see that $TS(\overline{E})$ and $RG(N)$ are isomorphic, as well as $DTMC(\overline{E})$ and $DTMC(N)$.

The following example shows that without the syntactic restriction on regularity of expressions the corresponding marked dts-boxes may be not safe.

Example 4.3. Let $E = [((\{a\},\frac{1}{2}) * ((\{b\},\frac{1}{2}) \| (\{c\},\frac{1}{2})) * (\{d\},\frac{1}{2})]$. In Figure 6, the marked dts-box $N = Box_{dts}(\overline{E})$ and its reachability graph $RG(N)$ are presented. In the marking $(0,1,1,2,0,0)$ there are 2 tokens in the place p_4. Symmetrically, in the marking

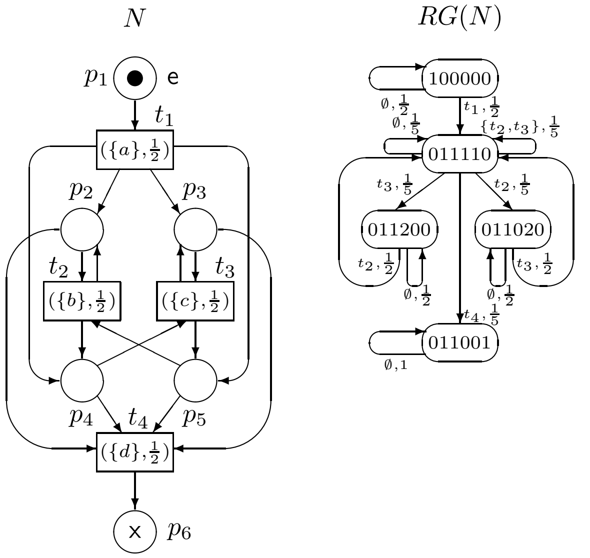

Fig. 6. The marked dts-box $N = Box_{dts}(\overline{E})$ for $E = [((\{a\},\frac{1}{2}) * ((\{b\},\frac{1}{2})\|(\{c\},\frac{1}{2}))*$ $(\{d\},\frac{1}{2})]$ and its reachability graph

$(0,1,1,0,2,0)$ there are 2 tokens in the place p_5. Thus, allowing concurrency in the second argument of iteration in the expression $\overline{E}$ can lead to non-safeness of the corresponding marked dts-box N, though, it is 2-bounded in the worst case (Best *et al.* 2001). The origin of the problem is that N has as a self-loop with two subnets which can function independently. Therefore, we have decided to consider regular expressions only, since the alternative, which is a safe version of the iteration operator with six arguments in the corresponding dts-box, like that from (Best *et al.* 2001), is rather cumbersome and has too intricate Petri net interpretation. Our motivation was to keep the algebraic and Petri net specifications as simple as possible.

5. Stochastic equivalences

Consider the expressions $E = (\{a\},\frac{1}{2})$ and $E' = (\{a\},\frac{1}{3})_1[](\{a\},\frac{1}{3})_2$, for which $\overline{E} \neq_{ts} \overline{E'}$, since $TS(\overline{E})$ has only one transition from the initial to the final state (with probability $\frac{1}{2}$) while $TS(\overline{E'})$ has two such ones (with probabilities $\frac{1}{4}$). On the other hand, all the mentioned transitions are labeled by activities with the same multiaction part $\{a\}$. Moreover, the overall probabilities of the mentioned transitions of $TS(\overline{E})$ and $TS(\overline{E'})$ coincide: $\frac{1}{2} = \frac{1}{4} + \frac{1}{4}$. Further, $TS(\overline{E})$ (as well as $TS(\overline{E'})$) has one empty loop transition from the initial state to itself with probability $\frac{1}{2}$ and one empty loop transition from the final state to itself with probability 1. The empty loop transitions are labeled by the empty multiset of activities. For calculating the transition probabilities of $TS(\overline{E'})$, take $\rho = \chi = \frac{1}{3}$ in Example 3.2. Unlike $=_{ts}$, most of the probabilistic and stochastic equivalences proposed in the literature do not differentiate between the processes such as those specified by E and E'.

Since the semantic equivalence $=_{ts}$ is too discriminating in many cases, we need weaker equivalence notions. These equivalences should possess the following necessary properties. First, any two equivalent processes must have the same sequences of multisets of multiactions, which are the multiaction parts of the activities executed in steps starting from the initial states of the processes. Second, for every such sequence, its execution probabilities within both processes must coincide. Third, the desired equivalence should preserve the branching structure of computations, i.e. the points of choice of an external observer between several extensions of a particular computation should be taken into account. In this section, we define two such notions: step stochastic bisimulation equivalence and stochastic isomorphism.

5.1. *Step stochastic bisimulation equivalence*

Bisimulation equivalences respect the particular points of choice in the behavior of a system. To define stochastic bisimulation equivalences, we have to consider a bisimulation as an *equivalence* relation that partitions the states of the *union* of the transition systems $TS(G)$ and $TS(G')$ of two dynamic expressions G and G' to be compared. For G and G' to be bisimulation equivalent, the initial states of their transition systems, $[G]_\approx$ and $[G']_\approx$, are to be related by a bisimulation having the following transfer property: two states are related if in each of them the same multisets of multiactions can occur, and the resulting states *belong to the same equivalence class*. In addition, the sums of probabilities for all such occurrences should be the same for both states.

Thus, we follow the approaches of (Jou and Smolka 1990; Larsen and Skou 1991; Hermanns and Rettelbach 1994; Hillston 1996; Bernardo and Gorrieri 1998; Bernardo 2007), but we implement step semantics instead of interleaving one considered in these papers. Recall also that we use the generative probabilistic transition systems, like in (Jou and Smolka 1990), in contrast to the reactive model, treated in (Larsen and Skou 1991), and we take transition probabilities instead of transition rates from (Hermanns and Rettelbach 1994; Hillston 1996; Bernardo and Gorrieri 1998; Bernardo 2007). Thus, step stochastic bisimulation equivalence that we define further is (in the probabilistic sense) comparable only with interleaving probabilistic bisimulation equivalence from (Jou and Smolka 1990), and our equivalence is obviously stronger.

In the definition below, we consider $\mathcal{L}(\Gamma) \in I\!N_f^{\mathcal{L}}$ for $\Gamma \in I\!N_f^{\mathcal{SL}}$, i.e. (possibly empty) multisets of multiactions. The multiactions can be empty, then $\mathcal{L}(\Gamma)$ contains the elements $\emptyset$, and it is not empty itself.

Let G be a dynamic expression and $\mathcal{H} \subseteq DR(G)$. Then, for any $s \in DR(G)$ and $A \in I\!N_f^{\mathcal{L}}$, we write $s \xrightarrow{A}_{\mathcal{P}} \mathcal{H}$, where $\mathcal{P} = PM_A(s, \mathcal{H})$ is the *overall probability to move from s into the set of states $\mathcal{H}$ via steps with the multiaction part A* defined as

$$PM_A(s, \mathcal{H}) = \sum_{\{\Gamma \mid \exists \tilde{s} \in \mathcal{H}\ s \xrightarrow{\Gamma} \tilde{s},\ \mathcal{L}(\Gamma) = A\}} PT(\Gamma, s).$$

We write $s \xrightarrow{A} \mathcal{H}$ if there exists $\mathcal{P}$ such that $s \xrightarrow{A}_{\mathcal{P}} \mathcal{H}$. Further, we write $s \rightarrow_{\mathcal{P}} \mathcal{H}$ if there exists A such that $s \xrightarrow{A} \mathcal{H}$, where $\mathcal{P} = PM(s, \mathcal{H})$ is the *overall probability to move from s into the set of states $\mathcal{H}$ via any steps* defined as

$$PM(s, \mathcal{H}) = \sum_{\{\Gamma \mid \exists \tilde{s} \in \mathcal{H}\ s \xrightarrow{\Gamma} \tilde{s}\}} PT(\Gamma, s).$$

To introduce a stochastic bisimulation between dynamic expressions G and G', we should consider the "composite" set of states $DR(G) \cup DR(G')$, since we have to identify the probabilities to come from any two equivalent states into the same "composite" equivalence class (w.r.t. the stochastic bisimulation). Note that, for $G \neq G'$, transitions starting from the states of $DR(G)$ (or $DR(G')$) always lead to those from the same set, since $DR(G) \cap DR(G') = \emptyset$, and this allows us to "mix" the sets of states in the definition of stochastic bisimulation.

Definition 5.1. Let G and G' be dynamic expressions. An *equivalence* relation $\mathcal{R} \subseteq (DR(G) \cup DR(G'))^2$ is a *step stochastic bisimulation* between G and G', denoted by $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$, if:

1 $([G]_{\approx}, [G']_{\approx}) \in \mathcal{R}$.
2 $(s_1, s_2) \in \mathcal{R} \ \Rightarrow\ \forall \mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}\ \forall A \in \mathbb{N}_f^{\mathcal{L}}$

$$s_1 \xrightarrow{A}_{\mathcal{P}} \mathcal{H} \ \Leftrightarrow\ s_2 \xrightarrow{A}_{\mathcal{P}} \mathcal{H}.$$

Dynamic expressions G and G' are *step stochastic bisimulation equivalent*, denoted by $G \underline{\leftrightarrow}_{ss} G'$, if there exists $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$.

Let $\mathcal{R}_{ss}(G, G') = \bigcup \{\mathcal{R} \mid \mathcal{R} : G \underline{\leftrightarrow}_{ss} G'\}$ be the *union of all step stochastic bisimulations* between G and G'. The following proposition proves that $\mathcal{R}_{ss}(G, G')$ is also an *equivalence* and $\mathcal{R}_{ss}(G, G') : G \underline{\leftrightarrow}_{ss} G'$.

Proposition 5.1. Let G and G' be dynamic expressions and $G \underline{\leftrightarrow}_{ss} G'$. Then $\mathcal{R}_{ss}(G, G')$ is the largest step stochastic bisimulation between G and G'.

Proof. See Appendix A.1. □

The algorithm for determining bisimulation of transition systems from (Paige and Tarjan 1987) can be adapted for our framework. This algorithm has time complexity $O(m \log n)$, where n is the number of states and m is the number of transitions.

5.2. *Stochastic isomorphism*

Stochastic isomorphism is weaker than $=_{ts}$. The main idea is to collect the probabilities of all transitions between the same pair of states such that the transition labels have the same multiaction parts.

For a dynamic expression G, let $s, \tilde{s} \in DR(G)$ and $s \xrightarrow{A}_{\mathcal{P}} \{\tilde{s}\}$. Then we write $s \xrightarrow{A}_{\mathcal{P}} \tilde{s}$.

Definition 5.2. Let G and G' be dynamic expressions. A mapping $\beta : DR(G) \to DR(G')$ is a *stochastic isomorphism* between G and G', denoted by $\beta : G =_{sto} G'$, if

1 β is a bijection such that $\beta([G]_\approx) = [G']_\approx$;

2 $\forall s, \tilde{s} \in DR(G)\ \forall A \in I\!N_f^{\mathcal{L}}\ s \xrightarrow{A}_{\mathcal{P}} \tilde{s} \ \Leftrightarrow\ \beta(s) \xrightarrow{A}_{\mathcal{P}} \beta(\tilde{s})$.

Dynamic expressions G and G' are *stochastically isomorphic*, denoted by $G =_{sto} G'$, if there exists $\beta : G =_{sto} G'$.

5.3. *Interrelations of the stochastic equivalences*

Now we compare the discrimination power of the stochastic equivalences.

Theorem 5.1. For dynamic expressions G and G' the following *strict* implications hold:

$$G \approx G' \ \Rightarrow\ G =_{ts} G' \ \Rightarrow\ G =_{sto} G' \ \Rightarrow\ G \underline{\leftrightarrow}_{ss} G'.$$

Proof. Let us check the validity of the forward implications.

— The implication $=_{sto} \to \underline{\leftrightarrow}_{ss}$ is proved as follows. Let $\beta : G =_{sto} G'$. Then it is easy to see that $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$, where $\mathcal{R} = \{(s, \beta(s)) \mid s \in DR(G)\}$.

— The implication $=_{ts} \to =_{sto}$ is valid, since stochastic isomorphism is that of transition systems up to merging of transitions with labels having identical multiaction parts.

— The implication $\approx \to =_{ts}$ is valid, since the transition system of a dynamic formula is defined based on its structural equivalence class.

Let us see that that the implications are strict, i.e. the reverse ones do not work, by the following counterexamples.

(a) Let $E = (\{a\}, \frac{1}{2}); (\{b\}, \frac{1}{2})$ and $E' = (\{a\}, \frac{1}{3}); (\{b\}, \frac{1}{2})[](\{a\}, \frac{1}{3}); (\{b\}, \frac{1}{2})$. Then $\overline{E} \underline{\leftrightarrow}_{ss} \overline{E'}$, but $\overline{E} \neq_{sto} \overline{E'}$, since $TS(\overline{E'})$ has more states than $TS(\overline{E})$.

(b) Let $E = (\{a\}, \frac{1}{2})$ and $E' = (\{a\}, \frac{1}{3})_1[](\{a\}, \frac{1}{3})_2$. Then $\overline{E} =_{sto} \overline{E'}$, but $\overline{E} \neq_{ts} \overline{E'}$, since $TS(\overline{E})$ has only one transition from the initial to the final state while $TS(\overline{E'})$ has two such ones.

(c) Let $E = (\{a\}, \frac{1}{2}); (\{\hat{a}\}, \frac{1}{2})$ and $E' = ((\{a\}, \frac{1}{2}); (\{\hat{a}\}, \frac{1}{2}))\ \mathsf{sy}\ a$. Then $\overline{E} =_{ts} \overline{E'}$, but $\overline{E} \not\approx \overline{E'}$, since $\overline{E}$ and $\overline{E'}$ cannot be reached from each other by inaction rules.

□

Example 5.1. In Figure 7, the marked dts-boxes corresponding to the dynamic expressions from equivalence examples of Theorem 5.1 are presented, i.e. $N = Box_{dts}(\overline{E})$ and $N' = Box_{dts}(\overline{E'})$ for each picture (a)–(c).

6. Reduction modulo equivalences

The equivalences which we proposed can be used to reduce transition systems and DTMCs of expressions (reachability graphs and DTMCs of dts-boxes). Reductions of graph-based models, like transition systems, reachability graphs and DTMCs, result in those with less states (the graph nodes). The goal of the reduction is to decrease the

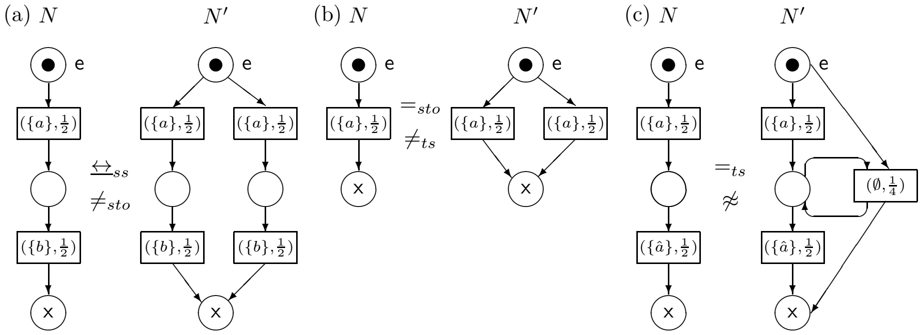

Fig. 7. Dts-boxes of the dynamic expressions from equivalence examples of Theorem 5.1

number of states in the semantic representation of the modeled system while preserving its important qualitative and quantitative properties. Thus, the reduction allows one to simplify the behaviour and performance analysis of systems.

An *autobisimulation* is a bisimulation between an expression and itself. For a dynamic expression G and a step stochastic autobisimulation on it $\mathcal{R} : G\underline{\leftrightarrow}_{ss}G$, let $\mathcal{K} \in DR(G)/_{\mathcal{R}}$ and $s_1, s_2 \in \mathcal{K}$. We have for all $\widetilde{\mathcal{K}} \in DR(G)/_{\mathcal{R}}$ and for all $A \in I\!N_f^{\mathcal{L}}$ the following holds: $s_1 \stackrel{A}{\to}_{\mathcal{P}} \widetilde{\mathcal{K}}$ iff $s_2 \stackrel{A}{\to}_{\mathcal{P}} \widetilde{\mathcal{K}}$. The previous statement is valid for all $s_1, s_2 \in \mathcal{K}$, hence, we can rewrite it as $\mathcal{K} \stackrel{A}{\to}_{\mathcal{P}} \widetilde{\mathcal{K}}$, where $\mathcal{P} = PM_A(\mathcal{K}, \widetilde{\mathcal{K}}) = PM_A(s_1, \widetilde{\mathcal{K}}) = PM_A(s_2, \widetilde{\mathcal{K}})$.

We write $\mathcal{K} \stackrel{A}{\to} \widetilde{\mathcal{K}}$ if there exists $\mathcal{P}$ such that $\mathcal{K} \stackrel{A}{\to}_{\mathcal{P}} \widetilde{\mathcal{K}}$ and $\mathcal{K} \to \widetilde{\mathcal{K}}$ if there exists A such that $\mathcal{K} \stackrel{A}{\to} \widetilde{\mathcal{K}}$ The similar arguments allow us to write $\mathcal{K} \to_{\mathcal{P}} \widetilde{\mathcal{K}}$, where $\mathcal{P} = PM(\mathcal{K}, \widetilde{\mathcal{K}}) = PM(s_1, \widetilde{\mathcal{K}}) = PM(s_2, \widetilde{\mathcal{K}})$.

The *average sojourn time in the equivalence class (w.r.t. $\mathcal{R}$) of states* $\mathcal{K}$ is $SJ_{\mathcal{R}}(\mathcal{K}) = \frac{1}{1-PM(\mathcal{K},\mathcal{K})}$. The *average sojourn time vector for the equivalence classes (w.r.t. $\mathcal{R}$) of states* of G, denoted by $SJ_{\mathcal{R}}$, has the elements $SJ_{\mathcal{R}}(\mathcal{K}),\ \mathcal{K} \in DR(G)/_{\mathcal{R}}$. The *sojourn time variance in the equivalence class (w.r.t. $\mathcal{R}$) of states* $\mathcal{K}$ is $VAR_{\mathcal{R}}(\mathcal{K}) = \frac{PM(\mathcal{K},\mathcal{K})}{(1-PM(\mathcal{K},\mathcal{K}))^2}$. The *sojourn time variance vector for the equivalence classes (w.r.t. $\mathcal{R}$) of states* of G, denoted by $VAR_{\mathcal{R}}$, has the elements $VAR_{\mathcal{R}}(\mathcal{K}),\ \mathcal{K} \in DR(G)/_{\mathcal{R}}$.

Let $\mathcal{R}_{ss}(G) = \bigcup\{\mathcal{R} \mid \mathcal{R} : G\underline{\leftrightarrow}_{ss}G\}$ be the *union of all step stochastic autobisimulations* on G. By Proposition 5.1, $\mathcal{R}_{ss}(G)$ is the largest step stochastic autobisimulation on G. Based on the equivalence classes w.r.t. $\mathcal{R}_{ss}(G)$, the quotient (by $\underline{\leftrightarrow}_{ss}$) transition systems and the quotient (by $\underline{\leftrightarrow}_{ss}$) underlying DTMCs of expressions can be defined. The mentioned equivalence classes become the quotient states. Every quotient transition between two such composite states represents all steps (having the same multiaction part in case of the transition system quotient) from the first state to the second one.

Definition 6.1. Let G be a dynamic expression. The *quotient (by $\underline{\leftrightarrow}_{ss}$) (labeled probabilistic) transition system* of G is a quadruple $TS_{\underline{\leftrightarrow}_{ss}}(G) = (S_{\underline{\leftrightarrow}_{ss}}, L_{\underline{\leftrightarrow}_{ss}}, \mathcal{T}_{\underline{\leftrightarrow}_{ss}}, s_{\underline{\leftrightarrow}_{ss}})$, where

— $S_{\underline{\leftrightarrow}_{ss}} = DR(G)/_{\mathcal{R}_{ss}(G)}$;
— $L_{\underline{\leftrightarrow}_{ss}} \subseteq I\!N_f^{\mathcal{L}} \times (0;1]$;
— $\mathcal{T}_{\underline{\leftrightarrow}_{ss}} = \{(\mathcal{K}, (A, PM_A(\mathcal{K}, \widetilde{\mathcal{K}})), \widetilde{\mathcal{K}}) \mid \mathcal{K}, \widetilde{\mathcal{K}} \in DR(G)/_{\mathcal{R}_{ss}(G)},\ \mathcal{K} \stackrel{A}{\to} \widetilde{\mathcal{K}}\}$;

— $s_{\underline{\leftrightarrow}_{ss}} = [[G]_{\approx}]_{\mathcal{R}_{ss}(G)}$.

The transition $(\mathcal{K}, (A, \mathcal{P}), \widetilde{\mathcal{K}}) \in \mathcal{T}_{\underline{\leftrightarrow}_{ss}}$ will be written as $\mathcal{K} \xrightarrow{A}_{\mathcal{P}} \widetilde{\mathcal{K}}$.

Definition 6.2. Let G be a dynamic expression. The *quotient (by $\underline{\leftrightarrow}_{ss}$) underlying DTMC* of G, denoted by $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$, has the state space $DR(G)/_{\mathcal{R}_{ss}(G)}$, the initial state $[[G]_{\approx}]_{\mathcal{R}_{ss}(G)}$ and the transitions $\mathcal{K} \rightarrow_{\mathcal{P}} \widetilde{\mathcal{K}}$, where $\mathcal{P} = PM(\mathcal{K}, \widetilde{\mathcal{K}})$.

The *quotient (by $\underline{\leftrightarrow}_{ss}$) average sojourn time vector* of G is $SJ_{\underline{\leftrightarrow}_{ss}} = SJ_{\mathcal{R}_{ss}(G)}$. The *quotient (by $\underline{\leftrightarrow}_{ss}$) sojourn time variance vector* of G is $VAR_{\underline{\leftrightarrow}_{ss}} = VAR_{\mathcal{R}_{ss}(G)}$.

The quotients of both transition systems and underlying DTMCs are the minimal reductions of the mentioned objects modulo $\underline{\leftrightarrow}_{ss}$. The quotients can be used to simplify analysis of system properties preserved by $\underline{\leftrightarrow}_{ss}$, since less states should be examined for it. Such reduction method resembles that from (Autant and Schnoebelen 1992) based on place bisimulation equivalence for PNs, excepting that the former method merges states, while the latter one merges places.

The algorithms which can be adapted for our framework exist for constructing the quotients of transition systems by bisimulation (Paige and Tarjan 1987) and those of (discrete or continuous time) Markov chains by ordinary lumping (Derisavi *et al.* 2003). The algorithms have time complexity $O(m \log n)$ and space complexity $O(m + n)$ (the case of Markov chains), where n is the number of states and m is the number of transitions. As mentioned in (Wimmer *et al.* 2010), the algorithm from (Derisavi *et al.* 2003) can be easily adjusted to produce quotients of labeled probabilistic transition systems by the probabilistic bisimulation equivalence. In (Wimmer *et al.* 2010), the symbolic partition refinement algorithm on state space of CTMCs was proposed. The algorithm can be straightforwardly accommodated to DTMCs and other Markovian models, Kripke structures and labeled probabilistic transition systems. Such a symbolic lumping uses memory efficiently due to compact representation of the state space partition. The symbolic lumping is time efficient, since fast algorithm of the partition representation and refinement is applied.

The comprehensive quotient example will be presented in Section 8.

7. Stationary behaviour

Let us examine how the proposed equivalences can be used to compare the behaviour of stochastic processes in their steady states. We shall consider only formulas specifying stochastic processes with infinite behavior, i.e. expressions with the iteration operator. Note that the iteration operator does not guarantee infiniteness of behaviour, since there can exist a deadlock within the body (the second argument) of iteration when the corresponding subprocess does not reach its final state by some reasons. Let us define the expression $\mathsf{Stop} = (\{g\}, \frac{1}{2})$ rs g specifying the non-terminating process that performs only empty loops with probability 1. In particular, if the body of iteration contains the Stop expression, then the iteration will be "broken". On the other hand, the iteration body can be left after a finite number of its repeated executions and then the iteration termination is started. To avoid executing any activities after the iteration body, we take Stop as the termination argument of iteration.

Like in the framework of DTMCs, in DTSPNs the most common systems for performance analysis are *ergodic* (recurrent non-null, aperiodic and irreducible) ones. For ergodic DTSPNs, the steady-state marking probabilities exist and can be determined. In (Molloy 1985), the following sufficient (but not necessary) conditions for ergodicity of DTSPNs are stated: *liveness* (for each transition and any reachable marking there exist a sequence of markings from it leading to the marking enabling that transition), *boundedness* (the number of tokens in every place is not greater than some fixed number for any reachable marking) and *nondeterminism* (the transition probabilities are strictly less than 1). Let the dts-box of a dynamic expression has no deadlocks in the body of some iteration operator it contains and Stop is the termination argument of this operator. Then the three ergodicity conditions are satisfied: the subnet corresponding to such an iteration body is live, safe (1-bounded) and nondeterministic (since all markings of the live subnet are non-terminal, the probabilities of transitions from them are strictly less than 1). Hence, its DTMC restricted to the states between the initial and final states of this iteration body is ergodic. The isomorphism between DTMCs of expressions and those of the corresponding dts-boxes which is stated by Proposition 4.1 guarantees that the underlying DTMC of an expression with infinite behaviour is ergodic if restricted to the states in which such an iteration body is executed.

In this section, we consider the expressions such that their underlying DTMCs contain one ergodic subset of states to guarantee that a single steady state exists.

7.1. *Theoretical background*

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

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

The transient (k-step, $k \in I\!N$) probability mass function (PMF) $\psi[k] = (\psi_1[k], \ldots, \psi_n[k])$ for $DTMC(G)$ is calculated as

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

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

Note also that $\psi[k+1] = \psi[k]\mathbf{P}$ $(k \in I\!N)$.

The steady-state PMF $\psi = (\psi_1, \ldots, \psi_n)$ for $DTMC(G)$ is a solution of the equation system

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

where $\mathbf{I}$ is the identity matrix of size n and $\mathbf{0}$ is a row vector of n values 0, $\mathbf{1}$ is that of n values 1.

If $DTMC(G)$ has a single steady state then $\psi = \lim_{k\to\infty} \psi[k]$.

For $s = s_i \in DR(G)$ $(1 \le i \le n)$ let $\psi[k](s) = \psi_i[k]$ $(k \in \mathbb{N})$ 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)* can be calculated based on the steady-state PMF for $DTMC(G)$.

— The *average recurrence (return) time in the state* s (i.e. 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 is $\sum_{s\in DR(G)} \psi(s)r(s)$.

7.2. *Steady state and equivalences*

The following proposition demonstrates that, for two dynamic expressions related by $\underline{\leftrightarrow}_{ss}$, the steady-state probabilities to enter into an equivalence class coincide. One can also interpret the result stating that the mean recurrence time for an equivalence class is the same for both expressions.

Proposition 7.1. Let G, G' be dynamic expressions with $\mathcal{R} : G\underline{\leftrightarrow}_{ss}G'$ and ψ be the steady-state PMF for $DTMC(G)$, ψ' be the steady-state PMF for $DTMC(G')$. Then for all $\mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ we have

$$\sum_{s\in\mathcal{H}\cap DR(G)} \psi(s) = \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'(s').$$

Proof. See Appendix A.2. □

Let G be a dynamic expression. The steady-state PMF $\psi_{\underline{\leftrightarrow}_{ss}}$ for $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$ is defined like the corresponding notion ψ for $DTMC(G)$. By Proposition 7.1, for all $\mathcal{H} \in DR(G)/_{\mathcal{R}_{ss}(G)}$ we have $\psi_{\underline{\leftrightarrow}_{ss}}(\mathcal{H}) = \sum_{s\in\mathcal{H}} \psi(s)$. Thus, for every equivalence class $\mathcal{H} \in DR(G)/_{\mathcal{R}_{ss}(G)}$, the value of $\psi_{\underline{\leftrightarrow}_{ss}}$ corresponding to $\mathcal{H}$ is the sum of all values of ψ corresponding to the states from $\mathcal{H}$. Hence, using $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$ instead of $DTMC(G)$ simplifies the analytical solution, since we have less states, but constructing the TPM for $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$, denoted by $\mathbf{P}_{\underline{\leftrightarrow}_{ss}}$, also requires some efforts, including determining $\mathcal{R}_{ss}(G)$ and calculating the probabilities to move from one equivalence class to other. The behaviour of $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$ stabilizes quicker than that of $DTMC(G)$ (if each of them has a single steady state), since $\mathbf{P}_{\underline{\leftrightarrow}_{ss}}$ is denser matrix than $\mathbf{P}$ due to the fact that the former matrix is smaller and the transitions between the equivalence classes "include" all the transitions between the states belonging to these equivalence classes.

By Proposition 7.1, $\underline{\leftrightarrow}_{ss}$ preserves the quantitative properties of the stationary behaviour. Now we intend to demonstrate that the qualitative properties of the stationary behaviour based on the multiaction labels are preserved as well.

Definition 7.1. A *derived step trace* of a dynamic expression G is a chain $\Sigma = A_1 \cdots A_n \in (I\!N_f^{\mathcal{L}})^*$, where there exists $s \in DR(G)$ such that $s \stackrel{\Gamma_1}{\to} s_1 \stackrel{\Gamma_2}{\to} \cdots \stackrel{\Gamma_n}{\to} s_n$ and $\mathcal{L}(\Gamma_i) = A_i$ $(1 \leq i \leq n)$. Then the *probability to execute the derived step trace* Σ *in* s is

$$PT(\Sigma, s) = \sum_{\{\Gamma_1,\ldots,\Gamma_n \mid s=s_0 \stackrel{\Gamma_1}{\to} s_1 \stackrel{\Gamma_2}{\to} \cdots \stackrel{\Gamma_n}{\to} s_n,\ \mathcal{L}(\Gamma_i)=A_i\ \ (1 \leq i \leq n)\}} \prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}).$$

The following theorem demonstrates that, for two dynamic expressions related by $\underline{\leftrightarrow}_{ss}$, the steady-state probabilities to enter into an equivalence class and start a derived step trace from it coincide.

Theorem 7.1. Let G, G' be dynamic expressions with $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$ and ψ be the steady-state PMF for $DTMC(G)$, ψ' be the steady-state PMF for $DTMC(G')$ and Σ be a derived step trace of G and G'. Then for all $\mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ we have

$$\sum_{s \in \mathcal{H} \cap DR(G)} \psi(s) PT(\Sigma, s) = \sum_{s' \in \mathcal{H} \cap DR(G')} \psi'(s') PT(\Sigma, s').$$

Proof. See Appendix A.3. $\square$

Example 7.1. Let $E = [(\{a\}, \frac{1}{2}) * ((\{b\}, \frac{1}{2}); ((\{c\}, \frac{1}{3})_1 [] (\{c\}, \frac{1}{3})_2)) * \mathsf{Stop}]$ and
$E' = [(\{a\}, \frac{1}{2}) * (((\{b\}, \frac{1}{3})_1; (\{c\}, \frac{1}{2})_1) [] ((\{b\}, \frac{1}{3})_2; (\{c\}, \frac{1}{2})_2)) * \mathsf{Stop}]$.
We have $\overline{E} =_{sto} \overline{E'}$, hence, $\overline{E} \underline{\leftrightarrow}_{ss} \overline{E'}$.

$DR(\overline{E})$ consists of the equivalence classes

$$\begin{aligned}
s_1 &= [[\overline{(\{a\}, \tfrac{1}{2})} * ((\{b\}, \tfrac{1}{2}); ((\{c\}, \tfrac{1}{3})_1 [] (\{c\}, \tfrac{1}{3})_2)) * \mathsf{Stop}]]_{\approx},\\
s_2 &= [[(\{a\}, \tfrac{1}{2}) * (\overline{(\{b\}, \tfrac{1}{2})}; ((\{c\}, \tfrac{1}{3})_1 [] (\{c\}, \tfrac{1}{3})_2)) * \mathsf{Stop}]]_{\approx},\\
s_3 &= [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); \overline{((\{c\}, \tfrac{1}{3})_1 [] (\{c\}, \tfrac{1}{3})_2)}) * \mathsf{Stop}]]_{\approx}.
\end{aligned}$$

$DR(\overline{E'})$ consists of the equivalence classes

$$\begin{aligned}
s_1' &= [[\overline{(\{a\}, \tfrac{1}{2})} * (((\{b\}, \tfrac{1}{3})_1; (\{c\}, \tfrac{1}{2})_1) [] ((\{b\}, \tfrac{1}{3})_2; (\{c\}, \tfrac{1}{2})_2)) * \mathsf{Stop}]]_{\approx},\\
s_2' &= [[(\{a\}, \tfrac{1}{2}) * \overline{(((\{b\}, \tfrac{1}{3})_1; (\{c\}, \tfrac{1}{2})_1) [] ((\{b\}, \tfrac{1}{3})_2; (\{c\}, \tfrac{1}{2})_2))} * \mathsf{Stop}]]_{\approx},\\
s_3' &= [[(\{a\}, \tfrac{1}{2}) * (((\{b\}, \tfrac{1}{3})_1; \overline{(\{c\}, \tfrac{1}{2})_1}) [] ((\{b\}, \tfrac{1}{3})_2; (\{c\}, \tfrac{1}{2})_2)) * \mathsf{Stop}]]_{\approx},\\
s_4' &= [[(\{a\}, \tfrac{1}{2}) * (((\{b\}, \tfrac{1}{3})_1; (\{c\}, \tfrac{1}{2})_1) [] ((\{b\}, \tfrac{1}{3})_2; \overline{(\{c\}, \tfrac{1}{2})_2})) * \mathsf{Stop}]]_{\approx}.
\end{aligned}$$

The steady-state PMFs ψ for $DTMC(\overline{E})$ and ψ' for $DTMC(\overline{E'})$ are

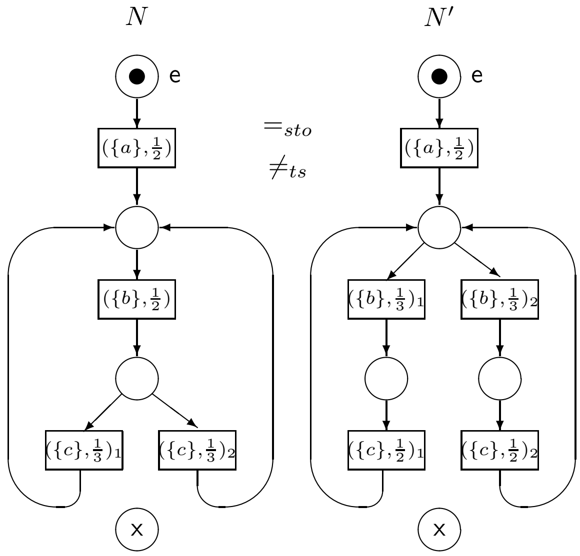

Fig. 8. $\underline{\leftrightarrow}_{ss}$ implies a coincidence of the steady-state probabilities to enter into an equivalence class and start a derived step trace from it

$$\psi = \left(0, \frac{1}{2}, \frac{1}{2}\right),\ \psi' = \left(0, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right).$$

Consider the equivalence class (w.r.t. $\mathcal{R}_{ss}(\overline{E}, \overline{E'})$) $\mathcal{H} = \{s_3, s'_3, s'_4\}$. One can see that the steady-state probabilities for $\mathcal{H}$ coincide: $\sum_{s\in\mathcal{H}\cap DR(\overline{E})} \psi(s) = \psi(s_3) = \frac{1}{2} = \frac{1}{4} + \frac{1}{4} = \psi'(s'_3) + \psi'(s'_4) = \sum_{s'\in\mathcal{H}\cap DR(\overline{E'})} \psi'(s')$. Let $\Sigma = \{\{c\}\}$. The steady-state probabilities to enter into the equivalence class $\mathcal{H}$ and start the derived step trace Σ from it coincide as well: $\psi(s_3)(PT(\{(\{c\}, \frac{1}{3})_1\}, s_3) + PT(\{(\{c\}, \frac{1}{3})_2\}, s_3)) = \frac{1}{2}\left(\frac{1}{4} + \frac{1}{4}\right) = \frac{1}{4} = \frac{1}{4} \cdot \frac{1}{2} + \frac{1}{4} \cdot \frac{1}{2} = \psi'(s'_3)PT(\{(\{c\}, \frac{1}{2})_1\}, s'_3) + \psi'(s'_4)PT(\{(\{c\}, \frac{1}{2})_2\}, s'_4)$.

In Figure 8, the marked dts-boxes corresponding to the dynamic expressions above are presented, i.e. $N = Box_{dts}(\overline{E})$ and $N' = Box_{dts}(\overline{E'})$.

7.3. *Preservation of performance and simplification of its analysis*

Many performance indices are based on the steady-state probabilities to enter into a set of similar states or, after coming in it, to start a derived step trace from this set. The similarity of states is usually captured by an equivalence relation, hence, the sets are often the equivalence classes. Proposition 7.1 and Theorem 7.1 guarantee a coincidence of the mentioned indices for the expressions related by $\underline{\leftrightarrow}_{ss}$. Thus, $\underline{\leftrightarrow}_{ss}$ (hence, all the stronger equivalences we have considered) preserves performance of stochastic systems modeled by expressions of dtsPBC.

In addition, it is easier to evaluate performance using a DTMC with less states, since in this case the size of the transition probability matrix is smaller, and we solve systems of less equations to calculate steady-state probabilities. The reasoning above validates the following method of performance analysis simplification.

1 The investigated system is specified by a static expression of dtsPBC.
2 The transition system of the expression is constructed.
3 After treating the transition system for self-similarity, a step stochastic autobisimulation equivalence for the expression is determined.
4 The quotient underlying DTMC is constructed from the quotient transition system.
5 Stationary probabilities and performance indices are calculated via the DTMC.

The limitation of the method above is its applicability only to the expressions such that their corresponding DTMCs contain one irreducible subset of states, i.e. the existence of exactly one stationary state is required. If a DTMC contains several irreducible subsets of states then several steady states may exist which depend on the initial PMF. There is an analytical method to determine the stable states for DTMCs of this kind as well (Kulkarni 2009). Note that, for every expression, the underlying DTMC has by definition only one initial PMF (that at the time moment 0), hence, the stationary state will be only one in this case too. The general steady-state probability will be calculated as a sum of the stationary probabilities of the irreducible subsets of states weighted by the probabilities to enter these subsets starting from the initial state and passing through some transient states. Further, it is worth applying the method only to the systems with similar subprocesses.

For transition systems reduction one can also use an analogue of the approach from (Katoen *et al.* 2011): first perform the fast symmetry reduction based on the method from (Kwiatkowska *et al.* 2006), then construct a quotient of the resulting transition system by bisimulation equivalence by applying the time-optimal partition refinement algorithm from (Derisavi *et al.* 2003) to the state space of this system. As mentioned in (Katoen *et al.* 2011), for a number of case studies, minimization by bisimulation results in more significant state space reduction than symmetry reduction, but the latter is much faster than the former, since symmetries are determined on a syntactical level. In (Baarir *et al.* 2011), the effective analysis methods were proposed for partially symmetric models.

8. Dining philosophers system

8.1. *The standard system*

Consider a model of five dining philosophers, for which the Petri net interpretation was proposed in (Peterson 1981). We investigate this dining philosophers system in the discrete time stochastic setting of dtsPBC. The philosophers occupy a round table, and there is one fork between every neighboring persons, hence, there are five forks on the table. A philosopher needs two forks to eat, namely, his left and right ones. Hence, all five philosophers cannot eat together, since otherwise there will not be enough forks available, but only one of two of them who are not neighbors. The model works as follows. After the activation of the system (the philosophers come in the dining room), five forks are placed on the table. If the left and right forks are available for a philosopher, he takes them simultaneously and begins eating. At the end of eating, the philosopher places both

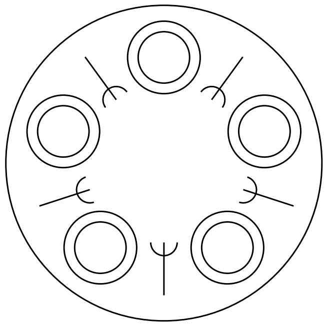

Fig. 9. The diagram of the dining philosophers system

his forks simultaneously back on the table. The strategy to pick up and release two forks simultaneously prevents the situation when a philosopher takes one fork but is not able to pick up the second one since their neighbor has already done so. In particular, we avoid a deadlock when all the philosophers take their left (right) forks and wait until their right (left) forks will be available. Figure 9 presents the diagram of the system.

One can explore what happens if there will be another number of philosophers at the table. The most interesting is to find the maximal sets of philosophers which can dine together, since all other combinations of the dining persons will be the subsets of these maximal sets. For the system with 1 philosopher the only maximal set is $\emptyset$. For the system with 2 philosophers the maximal sets are $\{1\},\ \{2\}$. For the system with 3 philosophers the maximal sets are $\{1\},\ \{2\},\ \{3\}$. For the system with 4 philosophers the maximal sets are $\{1,3\},\ \{2,4\}$. For the system with 5 philosophers the maximal sets are $\{1,3\},\ \{1,4\},\ \{2,4\},\ \{2,5\},\ \{3,5\}$. For the system with 6 philosophers the maximal sets are $\{1,4\},\ \{2,5\},\ \{3,6\},\ \{1,3,5\},\ \{2,4,6\}$. For the system with 7 philosophers the maximal sets are $\{1,3,5\},\ \{1,3,6\},\ \{1,4,6\},\ \{2,4,6\},\ \{2,4,7\},\ \{2,5,7\},\ \{3,5,7\}$. Thus, the system demonstrates a nontrivial behaviour when there are at least 5 philosophers.

Since the neighbors cannot dine together, the maximal number of the dining persons for the system with n philosophers will be $\lfloor\frac{n}{2}\rfloor$, i.e. the maximal natural number that is not greater than $\frac{n}{2}$. If the philosopher i belongs to some maximal set then the philosopher $i(\mathsf{mod}\ n)+1$ will belong to the next one. Let us calculate how many such maximal sets consisting of the maximal number of the philosophers ($\lfloor\frac{n}{2}\rfloor$) are there. If n is an even number then there will be only 2 such maximal sets of $\frac{n}{2}$ dining persons, namely, the philosophers numbered with all odd natural numbers which are not greater than n and those numbered with all even natural numbers which are not greater than n. If n is an odd number then there will be n such maximal sets of $\frac{n-1}{2}$ dining persons, since, starting from some maximal set one can "shift" clockwise $n-1$ times by one element modulo n until the next maximal set will coincide with the initial one.

We proceed with the 5 dining philosophers system. Let us explain the meaning of actions from the syntax of dtsPBC expressions which will specify the system modules. The action a corresponds to the system activation. The actions b_i and e_i correspond to the beginning and the end, respectively, of eating of philosopher i $(1\leq i\leq 5)$. The other actions are used for communication purposes only via synchronization, and we abstract from them later using restriction. Note that the expression of each philosopher includes two alternative subexpressions such that the second one specifies a resource (fork) sharing with the right neighbor.

The static expression of the philosopher i $(1 \leq i \leq 4)$ is
$E_i = [(\{x_i\}, \frac{1}{2}) * (((\{b_i, \widehat{y_i}\}, \frac{1}{2}); (\{e_i, \widehat{z_i}\}, \frac{1}{2}))[]((\{y_{i+1}\}, \frac{1}{2}); (\{z_{i+1}\}, \frac{1}{2}))) * \mathsf{Stop}]$.

The static expression of the philosopher 5 is
$E_5 = [(\{a, \widehat{x_1}, \widehat{x_2}, \widehat{x_2}, \widehat{x_4}\}, \frac{1}{2}) * (((\{b_5, \widehat{y_5}\}, \frac{1}{2}); (\{e_5, \widehat{z_5}\}, \frac{1}{2}))[]((\{y_1\}, \frac{1}{2}); (\{z_1\}, \frac{1}{2}))) * \mathsf{Stop}]$.

For $a_1, \ldots, a_n \in Act$ $(n \in I\!N)$, we shall abbreviate $\mathsf{sy}\ a_1 \cdots \mathsf{sy}\ a_n\ \mathsf{rs}\ a_1 \cdots \mathsf{rs}\ a_n$ to $\mathsf{sr}\ (a_1, \ldots, a_n)$. The static expression of the dining philosophers system is
$E = (E_1 \| E_2 \| E_3 \| E_4 \| E_5)\ \mathsf{sr}\ (x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4, y_5, z_1, z_2, z_3, z_4, z_5)$.

Let us illustrate an effect of synchronization. In the result of synchronization of the activities $(\{b_i, y_i\}, \frac{1}{2})$ and $(\{\widehat{y_i}\}, \frac{1}{2})$ we obtain the new activity $(\{b_i\}, \frac{1}{4})$ $(1 \leq i \leq 5)$. The synchronization of $(\{e_i, z_i\}, \frac{1}{2})$ and $(\{\widehat{z_i}\}, \frac{1}{2})$ produces $(\{e_i\}, \frac{1}{4})$ $(1 \leq i \leq 5)$. The result of synchronization of $(\{a, \widehat{x_1}, \widehat{x_2}, \widehat{x_3}, \widehat{x_4}\}, \frac{1}{2})$ and $(\{x_1\}, \frac{1}{2})$ is $(\{a, \widehat{x_2}, \widehat{x_3}, \widehat{x_4}\}, \frac{1}{4})$. The result of synchronization of $(\{a, \widehat{x_2}, \widehat{x_3}, \widehat{x_4}\}, \frac{1}{4})$ and $(\{x_2\}, \frac{1}{2})$ is $(\{a, \widehat{x_3}, \widehat{x_4}\}, \frac{1}{8})$. The result of synchronization of $(\{a, \widehat{x_3}, \widehat{x_4}\}, \frac{1}{8})$ and $(\{x_3\}, \frac{1}{2})$ is $(\{a, \widehat{x_4}\}, \frac{1}{16})$. The result of synchronization of $(\{a, \widehat{x_4}\}, \frac{1}{16})$ and $(\{x_4\}, \frac{1}{2})$ is $(\{a\}, \frac{1}{32})$.

$DR(\overline{E})$ has 12 states interpreted as follows: s_1 is the initial state, s_2: the system is activated and no philosophers dine, s_3: philosopher 1 dines, s_4: philosophers 1 and 4 dine, s_5: philosophers 1 and 3 dine, s_6: philosopher 4 dines, s_7: philosopher 3 dines, s_8: philosophers 2 and 4 dine, s_9: philosophers 3 and 5 dine, s_{10}: philosopher 2 dines, s_{11}: philosopher 5 dines, s_{12}: philosophers 2 and 5 dine.

In Figure 10, the transition system $TS(\overline{E})$ is presented.

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

$$SJ = \left(32, \frac{29}{20}, \frac{20}{11}, \frac{16}{7}, \frac{16}{7}, \frac{20}{11}, \frac{20}{11}, \frac{16}{7}, \frac{16}{7}, \frac{20}{11}, \frac{20}{11}, \frac{16}{7}\right).$$

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

$$VAR = \left(992, \frac{261}{400}, \frac{180}{121}, \frac{144}{49}, \frac{144}{49}, \frac{180}{121}, \frac{180}{121}, \frac{144}{49}, \frac{144}{49}, \frac{180}{121}, \frac{180}{121}, \frac{144}{49}\right).$$

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

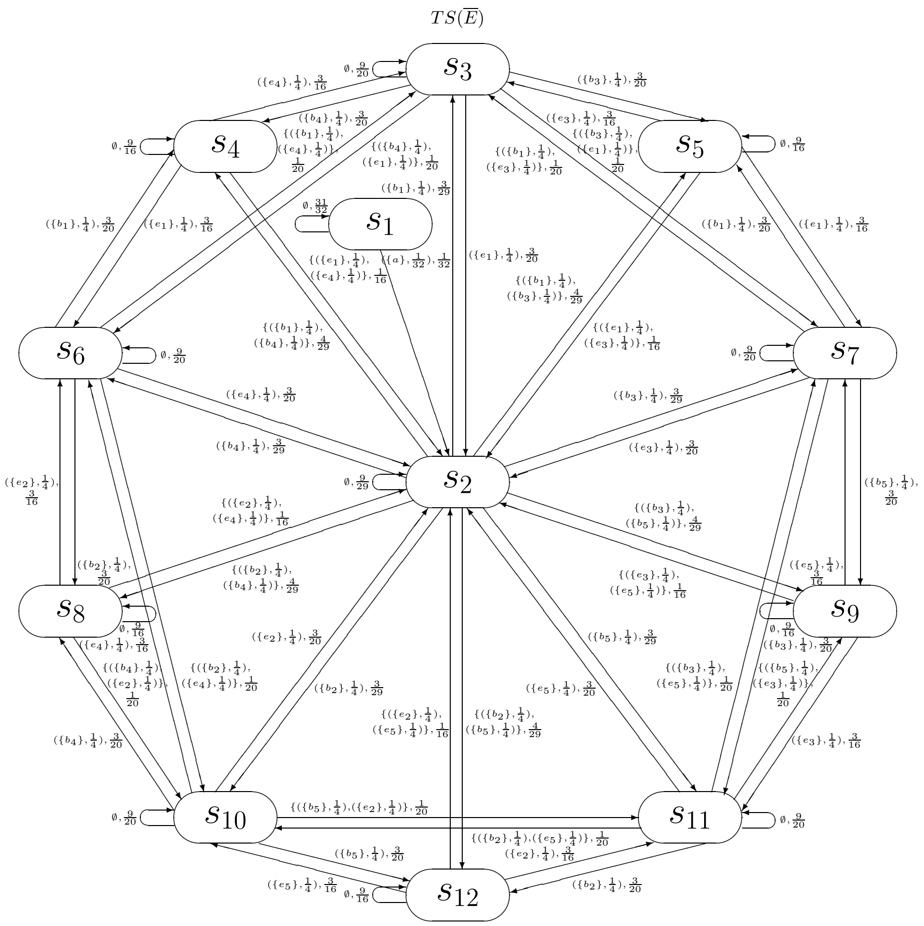

Fig. 10. The transition system of the dining philosophers system

$$\mathbf{P} = \begin{pmatrix}
\frac{31}{32} & \frac{1}{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{9}{29} & \frac{3}{29} & \frac{1}{29} & \frac{1}{29} & \frac{3}{29} & \frac{3}{29} & \frac{1}{29} & \frac{1}{29} & \frac{3}{29} & \frac{3}{29} & \frac{1}{29} \\
0 & \frac{3}{20} & \frac{9}{20} & \frac{3}{20} & \frac{3}{20} & \frac{1}{20} & \frac{1}{20} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{16} & \frac{3}{16} & \frac{9}{16} & 0 & \frac{3}{16} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{16} & \frac{3}{16} & 0 & \frac{9}{16} & 0 & \frac{3}{16} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{3}{20} & \frac{1}{20} & \frac{3}{20} & 0 & \frac{9}{20} & 0 & \frac{3}{20} & 0 & \frac{1}{20} & 0 & 0 \\
0 & \frac{3}{20} & \frac{1}{20} & 0 & \frac{3}{20} & 0 & \frac{9}{20} & 0 & \frac{3}{20} & 0 & \frac{1}{20} & 0 \\
0 & \frac{1}{16} & 0 & 0 & 0 & \frac{3}{16} & 0 & \frac{9}{16} & 0 & \frac{3}{16} & 0 & 0 \\
0 & \frac{1}{16} & 0 & 0 & 0 & 0 & \frac{3}{16} & 0 & \frac{9}{16} & 0 & \frac{3}{16} & 0 \\
0 & \frac{3}{20} & 0 & 0 & 0 & \frac{1}{20} & 0 & \frac{3}{20} & 0 & \frac{9}{20} & \frac{1}{20} & \frac{3}{20} \\
0 & \frac{3}{20} & 0 & 0 & 0 & 0 & \frac{1}{20} & 0 & \frac{3}{20} & \frac{1}{20} & \frac{9}{20} & \frac{3}{20} \\
0 & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{16} & \frac{3}{16} & \frac{9}{16}
\end{pmatrix}.$$

In Table 4, the transient and the steady-state probabilities $\psi_i[k]$ $(1 \leq i \leq 4)$ of the

Table 4. *Transient and steady-state probabilities of the dining philosophers system*

k	0	20	40	60	80	100	120	140	160	180	200	∞
$\psi_1[k]$	1	0.5299	0.2808	0.1488	0.0789	0.0418	0.0222	0.0117	0.0062	0.0033	0.0017	0
$\psi_2[k]$	0	0.0842	0.1098	0.1234	0.1306	0.1345	0.1365	0.1375	0.1381	0.1384	0.1386	0.1388
$\psi_3[k]$	0	0.0437	0.0681	0.0811	0.0880	0.0916	0.0935	0.0945	0.0951	0.0954	0.0955	0.0957
$\psi_4[k]$	0	0.0335	0.0537	0.0645	0.0701	0.0732	0.0748	0.0756	0.0760	0.0763	0.0764	0.0766

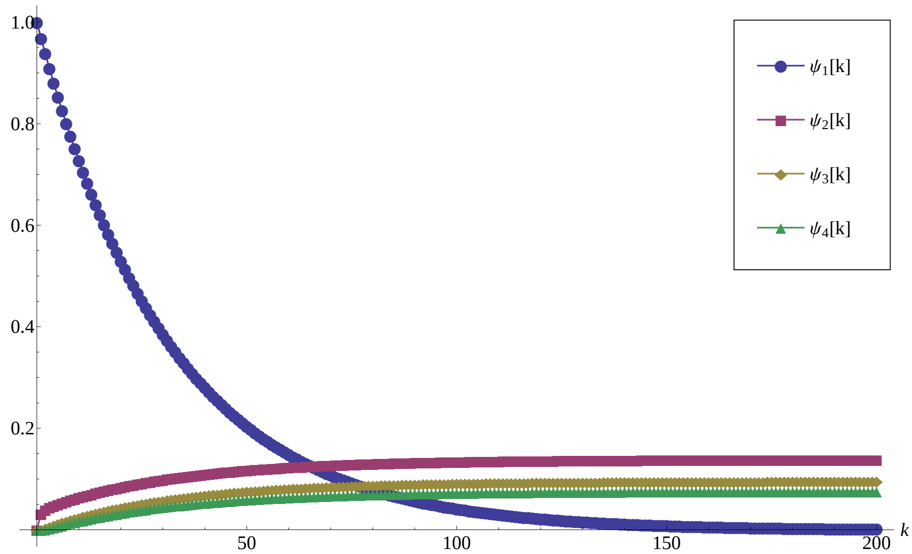

Fig. 11. Transient probabilities alteration diagram of the dining philosophers system

dining philosophers system at the time moments $k \in \{0, 20, 40, \ldots, 200\}$ and $k = \infty$ are presented, and in Figure 11, the alteration diagram (evolution in time) for the transient probabilities is depicted. It is sufficient to consider the probabilities for the states $s_1, \ldots, s_4$ only, since the corresponding values coincide for $s_3, s_6, s_7, s_{10}, s_{11}$ as well as for $s_4, s_5, s_8, s_9, s_{12}$.

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

$$\psi = \left(0, \frac{29}{209}, \frac{20}{209}, \frac{16}{209}, \frac{16}{209}, \frac{20}{209}, \frac{20}{209}, \frac{16}{209}, \frac{16}{209}, \frac{20}{209}, \frac{20}{209}, \frac{16}{209}\right).$$

We can now calculate the main performance indices.

— The average recurrence time in the state s_2, where all the forks are available, called the *average system run-through*, is $\frac{1}{\psi_2} = \frac{209}{29} = 7\frac{6}{29}$.

— Nobody eats in the state s_2. Then, the *fraction of time when no philosophers dine* is $\psi_2 = \frac{29}{209}$.

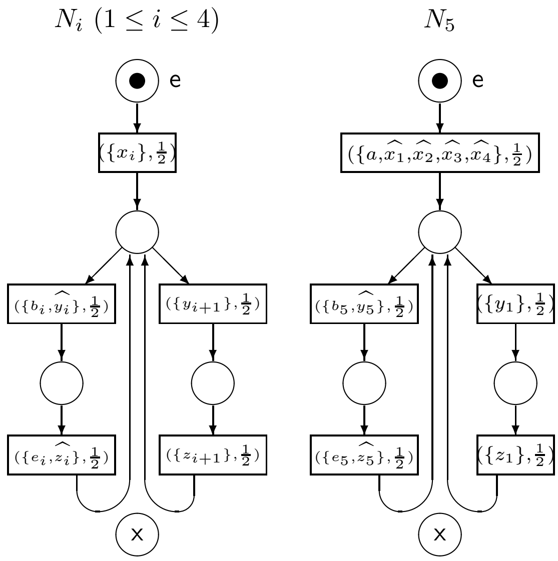

Fig. 12. The marked dts-boxes of the dining philosophers

Only one philosopher eats in the states $s_3, s_6, s_7, s_{10}, s_{11}$. Then, the *fraction of time when only one philosopher dines* is $\psi_3 + \psi_6 + \psi_7 + \psi_{10} + \psi_{11} = \frac{20}{209} + \frac{20}{209} + \frac{20}{209} + \frac{20}{209} + \frac{20}{209} = \frac{100}{209}$.
Two philosophers eat together in the states $s_4, s_5, s_8, s_9, s_{12}$. Then, the *fraction of time when two philosophers dine* is $\psi_4 + \psi_5 + \psi_8 + \psi_9 + \psi_{12} = \frac{16}{209} + \frac{16}{209} + \frac{16}{209} + \frac{16}{209} + \frac{16}{209} = \frac{80}{209}$.
The *relative fraction of time when two philosophers dine w.r.t. when only one philosopher dines* is $\frac{80}{209} \cdot \frac{209}{100} = \frac{4}{5}$.

— The beginning of eating of first philosopher $(\{b_1\}, \frac{1}{4})$ is only possible from the states s_2, s_6, s_7. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of activities containing $(\{b_1\}, \frac{1}{4})$. Thus, the *steady-state probability of the beginning of eating of first philosopher* is
$\psi_2 \sum_{\{\Gamma \mid (\{b_1\}, \frac{1}{4}) \in \Gamma\}} PT(\Gamma, s_2) + \psi_6 \sum_{\{\Gamma \mid (\{b_1\}, \frac{1}{4}) \in \Gamma\}} PT(\Gamma, s_6) + \psi_7 \sum_{\{\Gamma \mid (\{b_1\}, \frac{1}{4}) \in \Gamma\}} PT(\Gamma, s_7) = \frac{29}{209} \left(\frac{3}{29} + \frac{1}{29} + \frac{1}{29}\right) + \frac{20}{209} \left(\frac{3}{20} + \frac{1}{20}\right) + \frac{20}{209} \left(\frac{3}{20} + \frac{1}{20}\right) = \frac{13}{209}$.

In Figure 12, the marked dts-boxes corresponding to the dynamic expressions of the dining philosophers are presented, i.e. $N_i = Box_{dts}(\overline{E_i})$ $(1 \leq i \leq 5)$. In Figure 13, the marked dts-box corresponding to the dynamic expression of the dining philosophers system is depicted, i.e. $N = Box_{dts}(\overline{E})$.

8.2. *The abstract system and its reduction*

Let us consider a modification of the dining philosophers system with abstraction from personalities such that all the philosophers are indistinguishable. For example, we can just see that one or two philosophers dine but cannot observe who they are. We call this

N

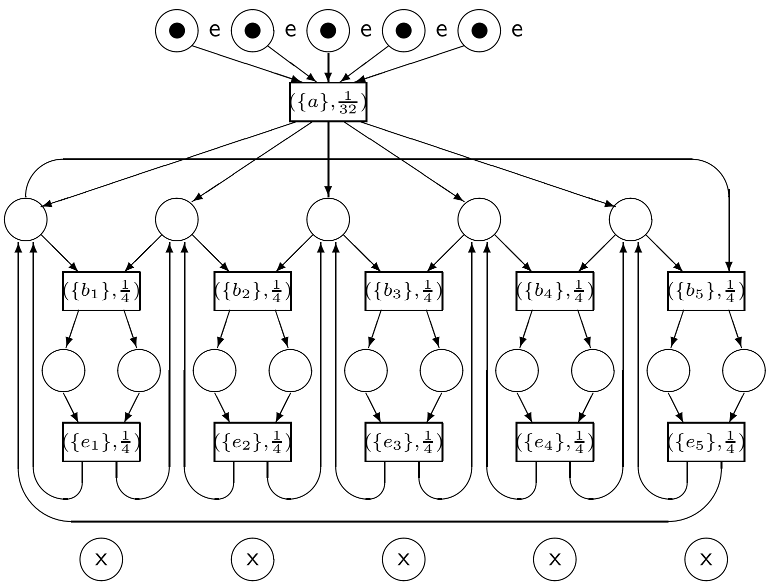

Fig. 13. The marked dts-box of the dining philosophers system

system the abstract dining philosophers one. To implement the abstraction, we replace the actions b_i and e_i $(1 \leq i \leq 5)$ in the system specification by b and e, respectively.

The static expression of the philosopher i $(1 \leq i \leq 4)$ is
$F_i = [(\{x_i\}, \frac{1}{2}) * (((\{b, \widehat{y_i}\}, \frac{1}{2}); (\{e, \widehat{z_i}\}, \frac{1}{2}))[]((\{y_{i+1}\}, \frac{1}{2}); (\{z_{i+1}\}, \frac{1}{2}))) * \mathsf{Stop}]$.

The static expression of the philosopher 5 is
$F_5 = [(\{a, \widehat{x_1}, \widehat{x_2}, \widehat{x_2}, \widehat{x_4}\}, \frac{1}{2}) * (((\{b, \widehat{y_5}\}, \frac{1}{2}); (\{e, \widehat{z_5}\}, \frac{1}{2}))[]((\{y_1\}, \frac{1}{2}); (\{z_1\}, \frac{1}{2}))) * \mathsf{Stop}]$.

The static expression of the abstract dining philosophers system is
$F = (F_1 \| F_2 \| F_3 \| F_4 \| F_5)$ sr $(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4, y_5, z_1, z_2, z_3, z_4, z_5)$.

$DR(\overline{F})$ resembles $DR(\overline{E})$, $TS(\overline{F})$ is similar to $TS(\overline{E})$ and $DTMC(\overline{F}) \simeq DTMC(\overline{E})$. Thus, the TPM and the steady-state PMF for $DTMC(\overline{F})$ and $DTMC(\overline{E})$ coincide.

The first performance index and the second group of them coincide for the standard and the abstract systems. The following performance index is based on non-personalized viewpoint to the philosophers.

— The beginning of eating of a philosopher $(\{b\}, \frac{1}{4})$ is only possible from the states $s_2, s_3, s_6, s_7, s_{10}, s_{11}$. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of activities containing $(\{b\}, \frac{1}{4})$. Thus, the *steady-state probability of the beginning of eating of a philosopher* is
$\psi_2 \sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}} PT(\Gamma, s_2) + \psi_3 \sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}} PT(\Gamma, s_3) +$
$\psi_6 \sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}} PT(\Gamma, s_6) + \psi_7 \sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}} PT(\Gamma, s_7) +$
$\psi_{10} \sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}} PT(\Gamma, s_{10}) + \psi_{11} \sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}} PT(\Gamma, s_{11}) =$
$\frac{29}{209}\left(\frac{3}{29} + \frac{1}{29} + \frac{3}{29} + \frac{1}{29} + \frac{3}{29} + \frac{1}{29} + \frac{3}{29} + \frac{1}{29} + \frac{3}{29} + \frac{1}{29}\right) +$

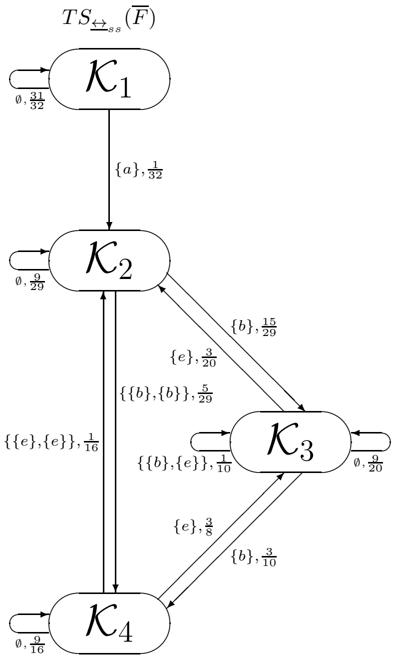

Fig. 14. The quotient transition system of the abstract dining philosophers system

$\frac{20}{209}\left(\frac{3}{20}+\frac{1}{20}+\frac{3}{20}+\frac{1}{20}\right)+\frac{20}{209}\left(\frac{3}{20}+\frac{1}{20}+\frac{3}{20}+\frac{1}{20}\right)+\frac{20}{209}\left(\frac{3}{20}+\frac{1}{20}+\frac{3}{20}+\frac{1}{20}\right)+$
$\frac{20}{209}\left(\frac{3}{20}+\frac{1}{20}+\frac{3}{20}+\frac{1}{20}\right)+\frac{20}{209}\left(\frac{3}{20}+\frac{1}{20}+\frac{3}{20}+\frac{1}{20}\right)=\frac{60}{209}$.

The marked dts-boxes corresponding to the dynamic expressions of the standard and the abstract dining philosophers are similar as well as the marked dts-boxes corresponding to the dynamic expression of the standard and the abstract dining philosophers systems.

We have $DR(\overline{F})/_{\mathcal{R}_{ss}(\overline{F})} = \{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_4\}$, where $\mathcal{K}_1 = \{s_1\}$ (the initial state), $\mathcal{K}_2 = \{s_2\}$ (the system is activated and no philosophers dine), $\mathcal{K}_3 = \{s_3, s_6, s_7, s_{10}, s_{11}\}$ (one philosopher dines), $\mathcal{K}_4 = \{s_4, s_5, s_8, s_9, s_{12}\}$ (two philosophers dine).

In Figure 14, the quotient transition system $TS_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is presented.

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

$$SJ' = \left(32, \frac{29}{20}, \frac{20}{9}, \frac{16}{7}\right).$$

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

$$VAR' = \left(992, \frac{261}{400}, \frac{180}{121}, \frac{144}{49}\right).$$

The TPM for $DTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is

Table 5. *Transient and steady-state probabilities of the quotient abstract dining philosophers system*

k	0	20	40	60	80	100	120	140	160	180	200	∞
$\psi'_1[k]$	1	0.5299	0.2808	0.1488	0.0789	0.0418	0.0222	0.0117	0.0062	0.0033	0.0017	0
$\psi'_2[k]$	0	0.0842	0.1098	0.1234	0.1306	0.1345	0.1365	0.1375	0.1381	0.1384	0.1386	0.1388
$\psi'_3[k]$	0	0.2183	0.3406	0.4054	0.4398	0.4580	0.4676	0.4727	0.4754	0.4769	0.4776	0.4785
$\psi'_4[k]$	0	0.1675	0.2687	0.3223	0.3507	0.3658	0.3738	0.3780	0.3802	0.3814	0.3821	0.3828

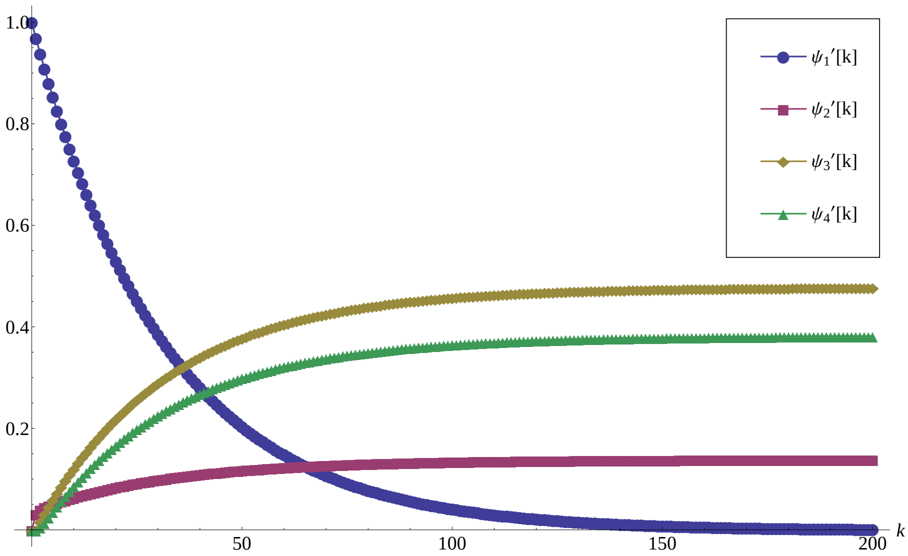

Fig. 15. Transient probabilities alteration diagram of the quotient abstract dining philosophers system

$$\mathbf{P}' = \begin{pmatrix} \frac{31}{32} & \frac{1}{32} & 0 & 0 \\ 0 & \frac{9}{29} & \frac{15}{29} & \frac{5}{29} \\ 0 & \frac{3}{20} & \frac{11}{20} & \frac{3}{10} \\ 0 & \frac{1}{16} & \frac{3}{8} & \frac{9}{16} \end{pmatrix}.$$

In Table 5, the transient and the steady-state probabilities $\psi'_i[k]$ $(1 \le i \le 4)$ of the quotient abstract dining philosophers system at the time moments $k \in \{0, 20, 40, \ldots, 200\}$ and $k = \infty$ are presented, and in Figure 15, the alteration diagram (evolution in time) for the transient probabilities is depicted.

The steady-state PMF for $DTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is

$$\psi' = \left(0, \frac{29}{209}, \frac{100}{209}, \frac{80}{209}\right).$$

We can now calculate the main performance indices.

— The average recurrence time in the state $\mathcal{K}_2$, where all the forks are available, called the *average system run-through*, is $\frac{1}{\psi'_2} = \frac{209}{29} = 7\frac{6}{29}$.

— Nobody eats in the state $\mathcal{K}_2$. The *fraction of time when no philosophers dine* is $\psi'_2 = \frac{29}{209}$.

Only one philosopher eats in the state $\mathcal{K}_3$. The *fraction of time when only one philosopher dines* is $\psi'_3 = \frac{100}{209}$.

Two philosophers eat together in the state $\mathcal{K}_4$. The *fraction of time when two philosophers dine* is $\psi'_4 = \frac{80}{209}$.

The *relative fraction of time when two philosophers dine w.r.t. when only one philosopher dines* is $\frac{80}{209} \cdot \frac{209}{100} = \frac{4}{5}$.

— The beginning of eating of a philosopher $\{b\}$ is only possible from the states $\mathcal{K}_2, \mathcal{K}_3$. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of multiactions containing $\{b\}$. Thus, the *steady-state probability of the beginning of eating of a philosopher* is
$\psi'_2 \sum_{\{A,\mathcal{K} \mid \{b\} \in A,\ \mathcal{K}_2 \xrightarrow{A} \mathcal{K}\}} PM_A(\mathcal{K}_2, \mathcal{K}) + \psi'_3 \sum_{\{A,\mathcal{K} \mid \{b\} \in A,\ \mathcal{K}_3 \xrightarrow{A} \mathcal{K}\}} PM_A(\mathcal{K}_3, \mathcal{K}) =$
$\frac{29}{209}\left(\frac{15}{29} + \frac{5}{29}\right) + \frac{100}{209}\left(\frac{3}{10} + \frac{1}{10}\right) = \frac{60}{209}$.

Observe that the performance indices are the same for the complete and the quotient abstract dining philosophers systems. The coincidence of the first performance index as well as the second group of indices obviously illustrates the result of Proposition 7.1. The coincidence of the third performance index is due to Theorem 7.1: one should just apply its result to the derived step traces $\{\{b\}\}$, $\{\{b\},\{b\}\}$, $\{\{b\},\{e\}\}$ of the expressions $\overline{F}$ and $\overline{F'}$, and then sum the left and right parts of the three resulting equalities.

8.3. *The generalized system*

Let us determine which is the influence of the multiaction probabilities from specification of the dining philosophers system on its performance. Let all these multiactions have the same probability ρ. The resulting specification K of the generalized dining philosophers system is defined as follows.

The static expression of the philosopher i ($1 \le i \le 4$) is
$K_i = [(\{x_i\}, \rho) * (((\{b_i, \widehat{y_i}\}, \rho); (\{e_i, \widehat{z_i}\}, \rho))[]((\{y_{i+1}\}, \rho); (\{z_{i+1}\}, \rho))) * \mathsf{Stop}]$.

The static expression of the philosopher 5 is
$K_5 = [(\{a, \widehat{x_1}, \widehat{x_2}, \widehat{x_2}, \widehat{x_4}\}, \rho) * (((\{b_5, \widehat{y_5}\}, \rho); (\{e_5, \widehat{z_5}\}, \rho))[]((\{y_1\}, \rho); (\{z_1\}, \rho))) * \mathsf{Stop}]$.

The static expression of the generalized dining philosophers system is
$K = (K_1 \| K_2 \| K_3 \| K_4 \| K_5)\ \mathsf{sr}\ (x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4, y_5, z_1, z_2, z_3, z_4, z_5)$.

$DR(\overline{K})$ has 12 states interpreted as follows: $\tilde{s}_1$ is the initial state, $\tilde{s}_2$: the system is activated and no philosophers dine, $\tilde{s}_3$: philosopher 1 dines, $\tilde{s}_4$: philosophers 1 and 4 dine, $\tilde{s}_5$: philosophers 1 and 3 dine, $\tilde{s}_6$: philosopher 4 dines, $\tilde{s}_7$: philosopher 3 dines, $\tilde{s}_8$: philosophers 2 and 4 dine, $\tilde{s}_9$: philosophers 3 and 5 dine, $\tilde{s}_{10}$: philosopher 2 dines, $\tilde{s}_{11}$: philosopher 5 dines, $\tilde{s}_{12}$: philosophers 2 and 5 dine.

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

$$\widetilde{SJ} = \left(\frac{1}{\rho^5}, \frac{1+3\rho^2+\rho^4}{5\rho^2}, \frac{1+\rho^2}{\rho^2(3-\rho^2)}, \frac{1}{\rho^2(2-\rho^2)}, \frac{1}{\rho^2(2-\rho^2)}, \frac{1+\rho^2}{\rho^2(3-\rho^2)},\right.$$
$$\left.\frac{1+\rho^2}{\rho^2(3-\rho^2)}, \frac{1}{\rho^2(2-\rho^2)}, \frac{1}{\rho^2(2-\rho^2)}, \frac{1+\rho^2}{\rho^2(3-\rho^2)}, \frac{1+\rho^2}{\rho^2(3-\rho^2)}, \frac{1}{\rho^2(2-\rho^2)}\right).$$

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

$$\widetilde{VAR} = \left(\frac{1-\rho^5}{\rho^{10}}, \frac{(1-\rho^2)^2(1+3\rho^2+\rho^4)}{25\rho^4}, \frac{(1-\rho^2)^2(1+\rho^2)}{\rho^4(3-\rho^2)^2}, \frac{(1-\rho^2)^2}{\rho^4(2-\rho^2)^2}, \frac{(1-\rho^2)^2}{\rho^4(2-\rho^2)^2}, \frac{(1-\rho^2)^2(1+\rho^2)}{\rho^4(3-\rho^2)^2},\right.$$
$$\left.\frac{(1-\rho^2)^2(1+\rho^2)}{\rho^4(3-\rho^2)^2}, \frac{(1-\rho^2)^2}{\rho^4(2-\rho^2)^2}, \frac{(1-\rho^2)^2}{\rho^4(2-\rho^2)^2}, \frac{(1-\rho^2)^2(1+\rho^2)}{\rho^4(3-\rho^2)^2}, \frac{(1-\rho^2)^2(1+\rho^2)}{\rho^4(3-\rho^2)^2}, \frac{(1-\rho^2)^2}{\rho^4(2-\rho^2)^2}\right).$$

Let us denote $\chi = 1-\rho^2$ and $\theta = 1+3\rho^2+\rho^4$. The TPM for $DTMC(\overline{K})$ is

$$\widetilde{\mathbf{P}} = \begin{pmatrix}
1-\rho^5 & \rho^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\chi^2}{\theta} & \frac{\rho^2\chi}{\theta} & \frac{\rho^4}{\theta} & \frac{\rho^4}{\theta} & \frac{\rho^2\chi}{\theta} & \frac{\rho^2\chi}{\theta} & \frac{\rho^4}{\theta} & \frac{\rho^4}{\theta} & \frac{\rho^2\chi}{\theta} & \frac{\rho^2\chi}{\theta} & \frac{\rho^4}{\theta} \\
0 & \frac{\rho^2\chi}{1+\rho^2} & \frac{\chi^2}{1+\rho^2} & \frac{\rho^2\chi}{1+\rho^2} & \frac{\rho^2\chi}{1+\rho^2} & \frac{\rho^4}{1+\rho^2} & \frac{\rho^4}{1+\rho^2} & 0 & 0 & 0 & 0 & 0 \\
0 & \rho^4 & \rho^2\chi & \chi^2 & 0 & \rho^2\chi & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \rho^4 & \rho^2\chi & 0 & \chi^2 & 0 & \rho^2\chi & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\rho^2\chi}{1+\rho^2} & \frac{\rho^4}{1+\rho^2} & \frac{\rho^2\chi}{1+\rho^2} & 0 & \frac{\chi^2}{1+\rho^2} & 0 & \frac{\rho^2\chi}{1+\rho^2} & 0 & \frac{\rho^4}{1+\rho^2} & 0 & 0 \\
0 & \frac{\rho^2\chi}{1+\rho^2} & \frac{\rho^4}{1+\rho^2} & 0 & \frac{\rho^2\chi}{1+\rho^2} & 0 & \frac{\chi^2}{1+\rho^2} & 0 & \frac{\rho^2\chi}{1+\rho^2} & 0 & \frac{\rho^4}{1+\rho^2} & 0 \\
0 & \rho^4 & 0 & 0 & 0 & \rho^2\chi & 0 & \chi^2 & 0 & \rho^2\chi & 0 & 0 \\
0 & \rho^4 & 0 & 0 & 0 & 0 & \rho^2\chi & 0 & \chi^2 & 0 & \rho^2\chi & 0 \\
0 & \frac{\rho^2\chi}{1+\rho^2} & 0 & 0 & 0 & \frac{\rho^4}{1+\rho^2} & 0 & \frac{\rho^2\chi}{1+\rho^2} & 0 & \frac{\chi^2}{1+\rho^2} & \frac{\rho^4}{1+\rho^2} & \frac{\rho^2\chi}{1+\rho^2} \\
0 & \frac{\rho^2\chi}{1+\rho^2} & 0 & 0 & 0 & 0 & \frac{\rho^4}{1+\rho^2} & 0 & \frac{\rho^2\chi}{1+\rho^2} & \frac{\rho^4}{1+\rho^2} & \frac{\chi^2}{1+\rho^2} & \frac{\rho^2\chi}{1+\rho^2} \\
0 & \rho^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \rho^2\chi & \rho^2\chi & \chi^2
\end{pmatrix}.$$

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

$$\tilde{\psi} = \frac{1}{11+8\rho^2+\rho^4}(0, 1+3\rho^2+\rho^4, 1+\rho^2, 1, 1, 1+\rho^2, 1+\rho^2, 1, 1, 1+\rho^2, 1+\rho^2, 1).$$

We can now calculate the main performance indices.

— The average recurrence time in the state s_2, where all the forks are available, called the *average system run-through*, is $\frac{1}{\tilde{\psi}_2} = \frac{11+8\rho^2+\rho^4}{1+3\rho^2+\rho^4}$.

— Nobody eats in the state s_2. The *fraction of time when no philosophers dine* is $\tilde{\psi}_2 = \frac{1+3\rho^2+\rho^4}{11+8\rho^2+\rho^4}$.

Only one philosopher eats in the states $s_3, s_6, s_7, s_{10}, s_{11}$. The *fraction of time when only one philosopher dines* is $\tilde{\psi}_3 + \tilde{\psi}_6 + \tilde{\psi}_7 + \tilde{\psi}_{10} + \tilde{\psi}_{11} = \frac{1+\rho^2}{11+8\rho^2+\rho^4} + \frac{1+\rho^2}{11+8\rho^2+\rho^4} + \frac{1+\rho^2}{11+8\rho^2+\rho^4} + \frac{1+\rho^2}{11+8\rho^2+\rho^4} + \frac{1+\rho^2}{11+8\rho^2+\rho^4} = \frac{5(1+\rho^2)}{11+8\rho^2+\rho^4}$.

Two philosophers eat together in the states $s_4, s_5, s_8, s_9, s_{12}$. The *fraction of time when two philosophers dine* is $\tilde{\psi}_4 + \tilde{\psi}_5 + \tilde{\psi}_8 + \tilde{\psi}_9 + \tilde{\psi}_{12} = \frac{1}{11+8\rho^2+\rho^4} + \frac{1}{11+8\rho^2+\rho^4} + \frac{1}{11+8\rho^2+\rho^4} + \frac{1}{11+8\rho^2+\rho^4} + \frac{1}{11+8\rho^2+\rho^4} = \frac{5}{11+8\rho^2+\rho^4}$.

The *relative fraction of time when two philosophers dine w.r.t. when only one philosopher dines* is $\frac{5}{11+8\rho^2+\rho^4} \cdot \frac{11+8\rho^2+\rho^4}{5(1+\rho^2)} = \frac{1}{1+\rho^2}$.

— The beginning of eating of first philosopher $(\{b_1\}, \rho^2)$ is only possible from the states s_2, s_6, s_7. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of activities containing $(\{b_1\}, \rho^2)$. Thus, the *steady-state probability of the beginning of eating of first philosopher* is

$\tilde{\psi}_2 \sum_{\{\Gamma|(\{b_1\},\rho^2)\in\Gamma\}} PT(\Gamma, s_2) + \tilde{\psi}_6 \sum_{\{\Gamma|(\{b_1\},\rho^2)\in\Gamma\}} PT(\Gamma, s_6) +$
$\tilde{\psi}_7 \sum_{\{\Gamma|(\{b_1\},\rho^2)\in\Gamma\}} PT(\Gamma, s_7) = \frac{1+3\rho^2+\rho^4}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} + \frac{\rho^4}{1+3\rho^2+\rho^4} + \frac{\rho^4}{1+3\rho^2+\rho^4}\right) +$
$\frac{1+\rho^2}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2}\right) + \frac{1+\rho^2}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2}\right) = \frac{\rho^2(3+\rho^2)}{11+8\rho^2+\rho^4}.$

8.4. *The abstract generalized system and its reduction*

Consider a modification of the generalized dining philosophers system with abstraction from personalities. We call this system the abstract generalized dining philosophers one.

The static expression of the philosopher i $(1 \le i \le 4)$ is
$L_i = [(\{x_i\}, \rho) * (((\{b, \widehat{y_i}\}, \rho); (\{e, \widehat{z_i}\}, \rho))[]\ ((\{y_{i+1}\}, \rho); (\{z_{i+1}\}, \rho))) * \mathsf{Stop}]$.

The static expression of the philosopher 5 is
$L_5 = [(\{a, \widehat{x_1}, \widehat{x_2}, \widehat{x_2}, \widehat{x_4}\}, \rho) * (((\{b, \widehat{y_5}\}, \rho); (\{e, \widehat{z_5}\}, \rho))[]((\{y_1\}, \rho); (\{z_1\}, \rho))) * \mathsf{Stop}]$.

The static expression of the abstract generalized dining philosophers system is
$L = (L_1 \| L_2 \| L_3 \| L_4 \| L_5)\ \mathsf{sr}\ (x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4, y_5, z_1, z_2, z_3, z_4, z_5)$.

$DR(\overline{L})$ resembles $DR(\overline{K})$, and $TS(\overline{L})$ is similar to $TS(\overline{K})$. We have $DTMC(\overline{L}) \simeq DTMC(\overline{K})$. Thus, the TPM and the steady-state PMF for $DTMC(\overline{L})$ and $DTMC(\overline{K})$ coincide.

The first performance index and the second group of the indices coincide for the standard and the abstract generalized systems. The following performance index is based on non-personalized viewpoint to the philosophers.

— The beginning of eating of a philosopher $(\{b\}, \rho^2)$ is only possible from the states $\tilde{s}_2, \tilde{s}_3, \tilde{s}_6, \tilde{s}_7, \tilde{s}_{10}, \tilde{s}_{11}$. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of activities containing $(\{b\}, \rho^2)$. Thus, the *steady-state probability of the beginning of eating of a philosopher* is
$\tilde{\psi}_2 \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT(\Gamma, \tilde{s}_2) + \tilde{\psi}_3 \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT(\Gamma, \tilde{s}_3) +$
$\tilde{\psi}_6 \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT(\Gamma, \tilde{s}_6) + \tilde{\psi}_7 \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT(\Gamma, \tilde{s}_7) +$
$\tilde{\psi}_{10} \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT(\Gamma, \tilde{s}_{10}) + \tilde{\psi}_{11} \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT(\Gamma, \tilde{s}_{11}) =$
$\frac{1+3\rho^2+\rho^4}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} + \frac{\rho^4}{1+3\rho^2+\rho^4} + \frac{\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} + \frac{\rho^4}{1+3\rho^2+\rho^4} + \frac{\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} +\right.$
$\left.\frac{\rho^4}{1+3\rho^2+\rho^4} + \frac{\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} + \frac{\rho^4}{1+3\rho^2+\rho^4} + \frac{\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} + \frac{\rho^4}{1+3\rho^2+\rho^4}\right) +$
$\frac{1+\rho^2}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2} + \frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2}\right) +$
$\frac{1+\rho^2}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2} + \frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2}\right) +$
$\frac{1+\rho^2}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2} + \frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2}\right) +$
$\frac{1+\rho^2}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2} + \frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2}\right) +$
$\frac{1+\rho^2}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2} + \frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2}\right) = \frac{15\rho^2}{11+8\rho^2+\rho^4}.$

We have $DR(\overline{L})/_{\mathcal{R}_{ss}(\overline{L})} = \{\widetilde{\mathcal{K}}_1, \widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}_3, \widetilde{\mathcal{K}}_4\}$, where $\widetilde{\mathcal{K}}_1 = \{\tilde{s}_1\}$ (the initial state), $\widetilde{\mathcal{K}}_2 = \{\tilde{s}_2\}$ (the system is activated and no philosophers dine), $\widetilde{\mathcal{K}}_3 = \{\tilde{s}_3, \tilde{s}_6, \tilde{s}_7, \tilde{s}_{10}, \tilde{s}_{11}\}$ (one philosopher dines), $\widetilde{\mathcal{K}}_4 = \{\tilde{s}_4, \tilde{s}_5, \tilde{s}_8, \tilde{s}_9, \tilde{s}_{12}\}$ (two philosophers dine).

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

$$\widetilde{SJ}' = \left(\frac{1}{\rho^5}, \frac{1+3\rho^2+\rho^4}{5\rho^2}, \frac{1+\rho^2}{\rho^2(3-\rho^2)}, \frac{1}{\rho^2(2-\rho^2)}\right).$$

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

$$\widetilde{VAR}' = \left(\frac{1-\rho^5}{\rho^{10}}, \frac{(1-\rho^2)^2(1+3\rho^2+\rho^4)}{25\rho^4}, \frac{(1-\rho^2)^2(1+\rho^2)}{\rho^4(3-\rho^2)^2}, \frac{(1-\rho^2)^2}{\rho^4(2-\rho^2)^2}\right).$$

The TPM for $DTMC_{\underline{\leftrightarrow}_{ss}}(\overline{L})$ is

$$\widetilde{\mathbf{P}}' = \begin{pmatrix} 1-\rho^5 & \rho^5 & 0 & 0 \\ 0 & \frac{(1-\rho^2)^2}{1+3\rho^2+\rho^4} & \frac{5\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} & \frac{5\rho^4}{1+3\rho^2+\rho^4} \\ 0 & \frac{\rho^2(1-\rho^2)}{1+\rho^2} & \frac{1-2\rho^2+3\rho^4}{1+\rho^2} & \frac{2\rho^2(1-\rho^2)}{1+\rho^2} \\ 0 & \rho^4 & 2\rho^2(1-\rho^2) & (1-\rho^2)^2 \end{pmatrix}.$$

The steady-state PMF for $DTMC_{\underline{\leftrightarrow}_{ss}}(\overline{L})$ is

$$\tilde{\psi}' = \left(0, \frac{1+3\rho^2+\rho^4}{11+8\rho^2+\rho^4}, \frac{5(1+\rho^2)}{11+8\rho^2+\rho^4}, \frac{5}{11+8\rho^2+\rho^4}\right).$$

We can now calculate the main performance indices.

- The average recurrence time in the state $\widetilde{\mathcal{K}}_2$, where all the forks are available, called the *average system run-through*, is $\frac{1}{\tilde{\psi}'_2} = \frac{11+8\rho^2+\rho^4}{1+3\rho^2+\rho^4}$.
- Nobody eats in the state $\widetilde{\mathcal{K}}_2$. The *fraction of time when no philosophers dine* is $\tilde{\psi}'_2 = \frac{1+3\rho^2+\rho^4}{11+8\rho^2+\rho^4}$.

  Only one philosopher eats in the state $\widetilde{\mathcal{K}}_3$. The *fraction of time when only one philosopher dines* is $\tilde{\psi}'_3 = \frac{5(1+\rho^2)}{11+8\rho^2+\rho^4}$.

  Two philosophers eat together in the state $\widetilde{\mathcal{K}}_4$. The *fraction of time when two philosophers dine* is $\tilde{\psi}'_4 = \frac{5}{11+8\rho^2+\rho^4}$.

  The *relative fraction of time when two philosophers dine w.r.t. when only one philosopher dines* is $\frac{5}{11+8\rho^2+\rho^4} \cdot \frac{11+8\rho^2+\rho^4}{5(1+\rho^2)} = \frac{1}{1+\rho^2}$.
- The beginning of eating of a philosopher $\{b\}$ is only possible from the states $\widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}_3$. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of multiactions containing $\{b\}$. Thus, the *steady-state probability of the beginning of eating of a philosopher* is
  $\tilde{\psi}'_2 \sum_{\{A,\widetilde{\mathcal{K}} \mid \{b\}\in A,\ \widetilde{\mathcal{K}}_2 \xrightarrow{A} \widetilde{\mathcal{K}}\}} PM_A(\widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}) + \tilde{\psi}'_3 \sum_{\{A,\widetilde{\mathcal{K}} \mid \{b\}\in A,\ \widetilde{\mathcal{K}}_3 \xrightarrow{A} \widetilde{\mathcal{K}}\}} PM_A(\widetilde{\mathcal{K}}_3, \widetilde{\mathcal{K}}) =$
  $\frac{1+3\rho^2+\rho^4}{11+8\rho^2+\rho^4}\left(\frac{5\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} + \frac{5\rho^4}{1+3\rho^2+\rho^4}\right) + \frac{5(1+\rho^2)}{11+8\rho^2+\rho^4}\left(\frac{2\rho^2(1-\rho^2)}{1+\rho^2} + \frac{2\rho^4}{1+\rho^2}\right) = \frac{15\rho^4}{11+8\rho^2+\rho^4}$.

Observe again that the performance indices are the same for the complete and the quotient abstract generalized dining philosophers systems. The explanation of this fact is just the same as that presented earlier for the complete and the quotient abstract dining philosophers systems.

Let us consider what is the effect of quantitative changes of the parameter ρ upon performance of the quotient abstract generalized dining philosophers system in its steady

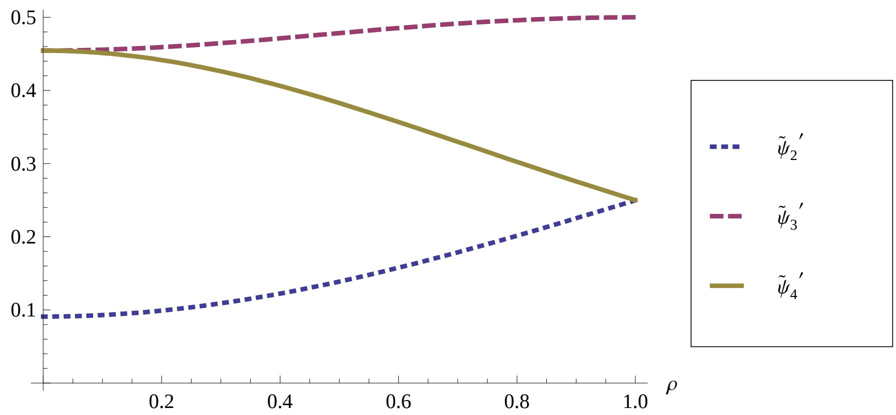

Fig. 16. Steady-state probabilities $\tilde{\psi}'_2,\ \tilde{\psi}'_3,\ \tilde{\psi}'_4$ as functions of the parameter ρ

state. Remember that $\rho \in (0;1)$ is the probability of every multiaction of the system. The closer is ρ to 0, the less is the probability to execute some activities at every discrete time step, hence, the system will most probably *stand idle*. The closer is ρ to 1, the greater is the probability to execute some activities at every discrete time step, hence, the system will most probably *operate*.

Since $\tilde{\psi}'_1 = 0$, only $\tilde{\psi}'_2 = \frac{1+3\rho^2+\rho^4}{11+8\rho^2+\rho^4},\ \tilde{\psi}'_3 = \frac{5(1+\rho^2)}{11+8\rho^2+\rho^4},\ \tilde{\psi}'_4 = \frac{5}{11+8\rho^2+\rho^4}$ depend on ρ. In Figure 16, the graphs of $\tilde{\psi}'_2,\ \tilde{\psi}'_3,\ \tilde{\psi}'_4$ as functions of ρ are depicted. The diagrams for $\tilde{\psi}'_2$ and $\tilde{\psi}'_4$ are symmetric w.r.t. the constant probability $\frac{1}{4}$. One can see that, the more is value of ρ, the less is difference between $\tilde{\psi}'_2$ and $\tilde{\psi}'_4$ and the more is difference between $\tilde{\psi}'_3$ and $\tilde{\psi}'_4$. Notice that, however, we do not allow $\rho = 0$ or $\rho = 1$.

In Figure 17, the plot of the average system run-through, calculated as $\frac{1}{\tilde{\psi}'_2}$, as a function of ρ is depicted. One can see that the run-through tends to 11 when ρ approaches 0, whereas it tends to 4 when ρ approaches 1. To speed up the operation of the system, one should take the parameter ρ closer to 1.

The fraction of time when no philosophers dine, calculated as $\tilde{\psi}'_2$, tends to $\frac{1}{11}$ when ρ approaches 0, whereas it tends to $\frac{1}{4}$ when ρ approaches 1. The fraction of time when only one philosopher dines, calculated as $\tilde{\psi}'_3$, tends to $\frac{5}{11}$ when ρ approaches 0, whereas it tends to $\frac{1}{2}$ when ρ approaches 1. The fraction of time when two philosophers dine, calculated as $\tilde{\psi}'_4$, tends to $\frac{5}{11}$ when ρ approaches 0, whereas it tends to $\frac{1}{4}$ when ρ approaches 1.

The first plot in Figure 18 represents the relative fraction of time when two philosophers dine w.r.t. when only one philosopher dines, calculated as $\frac{\tilde{\psi}'_4}{\tilde{\psi}'_3}$, as a function of ρ. One can see that the relative fraction tends to 1 when ρ approaches 0, whereas it tends to $\frac{1}{2}$ when ρ approaches 1. To increase the mentioned relative fraction, one should take the parameter ρ closer to 0.

The second plot in Figure 18 represents the steady-state probability of the beginning of eating of a philosopher, calculated as $\tilde{\psi}'_2\widetilde{\Sigma}'_2 + \tilde{\psi}'_3\widetilde{\Sigma}'_3$, where $\widetilde{\Sigma}'_i = \sum_{\{A,\widetilde{\mathcal{K}} \mid \{b\}\in A,\ \widetilde{\mathcal{K}}_i \xrightarrow{A} \widetilde{\mathcal{K}}\}} PM_A(\widetilde{\mathcal{K}}_i, \widetilde{\mathcal{K}}),\ i \in \{2,3\}$, as a function of ρ. One can see that

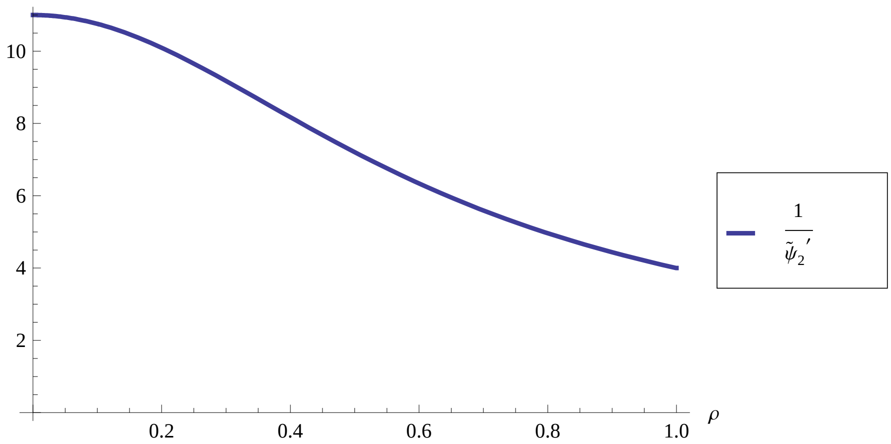

Fig. 17. Average system run-through $\frac{1}{\tilde{\psi}'_2}$ as a function of the parameter ρ

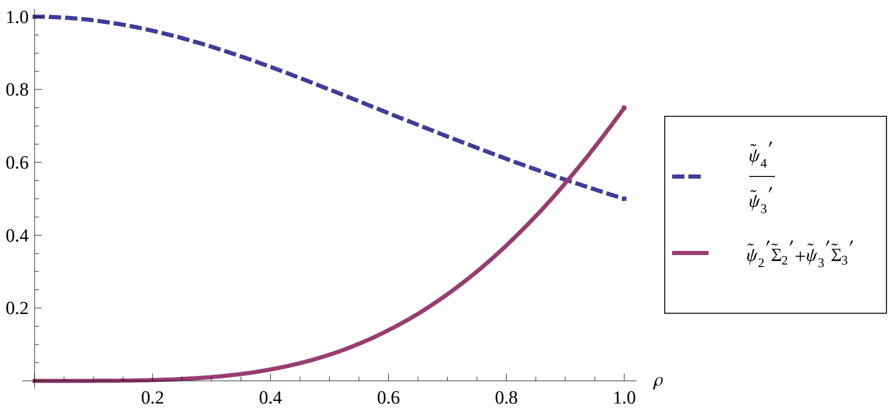

Fig. 18. Some performance indices as functions of the parameter ρ

the probability tends to 0 when ρ approaches 0 and it tends to $\frac{3}{4}$, when ρ approaches 1. To increase the mentioned probability, one should take the parameter ρ closer to 1.

Since $\tilde{\psi}'_4 - \tilde{\psi}'_2 = \frac{4-3\rho^2-\rho^4}{11+8\rho^2+\rho^4}$, the difference tends to $\frac{4}{11}$ when ρ approaches 0, whereas it tends to 0 when ρ approaches 1. The difference can be treated as that between the fractions of time when two and when no philosophers dine. Since $\tilde{\psi}'_3 - \tilde{\psi}'_4 = \frac{5\rho^2}{11+8\rho^2+\rho^4}$, the difference tends to 0 when ρ approaches 0, whereas it tends to $\frac{1}{4}$ when ρ approaches 1. The difference can be treated as the that between the fractions of time when one and when two philosophers dine.

From the performance viewpoint, it is more interesting is to consider the expression $\tilde{\psi}'_3 + \tilde{\psi}'_4 - \tilde{\psi}'_2 = \frac{9+2\rho^2-\rho^4}{11+8\rho^2+\rho^4}$. In Figure 19, the graph of $\tilde{\psi}'_3 + \tilde{\psi}'_4 - \tilde{\psi}'_2$ as a function of ρ is depicted. The value of the expression tends to $\frac{9}{11}$ when ρ approaches 0, whereas it tends

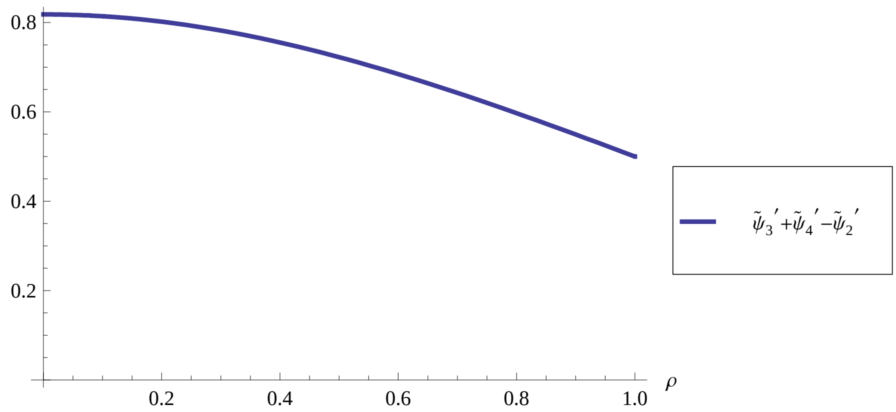

Fig. 19. Expression $\tilde{\psi}'_3 + \tilde{\psi}'_4 - \tilde{\psi}'_2$ as a function of the parameter ρ

to $\frac{1}{2}$ when ρ approaches 1. The value can be interpreted as the difference between the fractions of time when some (one or two) and when no philosophers dine.

Thus, when ρ is closer to 0, more time is spent for eating and less time remains for thinking, i.e. *dining is preferred*. In this case, the dining time fractions for one and two philosophers approach the same value $\frac{5}{11}$ (the relative time fraction tends to 1). When ρ is closer to 1, the situation is symmetric, i.e. *thinking is preferred*. In this case, the dining time fraction of one philosopher approaches its maximum $\frac{1}{2}$, whereas the dining time fraction of two philosophers approaches its minimum $\frac{1}{4}$ (the relative time fraction tends to $\frac{1}{2}$).

9. Related work

In this section, we consider in detail differences and similarities between dtsPBC and other well-known or similar SPAs for the purpose of subsequent determining the specific advantages of dtsPBC.

9.1. *Continuous time and interleaving semantics*

Let us compare dtsPBC with classical SPAs: Markovian Timed Processes for Performance Evaluation (MTIPP) (Hermanns and Rettelbach 1994), Performance Evaluation Process Algebra (PEPA) (Hillston 1996) and Extended Markovian Process Algebra (EMPA) (Bernardo and Gorrieri 1998).

In MTIPP, every activity is a pair consisting of the action name (including the symbol τ for the *internal*, invisible action) and the parameter of exponential distribution of the activity duration (the *rate*). The operations are *prefix*, *choice*, *parallel* composition including *synchronization* on the specified action set and *recursion*. It is possible to specify processes by recursive equations as well. The interleaving semantics is defined on the basis of Markovian (i.e. extended with the specification of rates) labeled transition systems. Note that we have the interleaving behaviour here because the exponential probability distribution function is a continuous one, and a simultaneous firing of any two activities has zero probability according to the properties of continuous distributions. The continuous time Markov chains (CTMCs) can be derived from the mentioned transition systems to analyze the performance.

In PEPA, activities are the pairs consisting of action types (including the *unknown*, unimportant type τ) and activity rates. The rate is either the parameter of exponential distribution of the activity duration or it is *unspecified*, denoted by $\top$. An activity with unspecified rate is *passive* by its action type. The set of operations includes *prefix*, *choice*, *cooperation*, *hiding* and constants whose meaning is given by the defining equations including the *recursive* ones. The cooperation is accomplished on the set of action types (the cooperation set) on which the components must *synchronize* or cooperate. If the cooperation set is empty, the cooperation operator turns into the *parallel* combinator. The semantics is interleaving, it is defined via the extension of labeled transition systems with a possibility to specify activity rates. Based on the transition systems, the continuous time Markov processes (CTMPs) are generated which are used for performance evaluation with the help of the embedded continuous time Markov chains (ECTMCs).

In EMPA, each action is a pair consisting of its type and rate. Actions can be *external* or *internal* (denoted by τ) according to types. There are three kinds of actions according to rates: *timed* ones with exponentially distributed durations (essentially, the actions from MTIPP and PEPA), *immediate* ones with priorities and weights (the actions analogous to immediate transitions of generalized SPNs, GSPNs) and *passive* ones (similar to passive actions of PEPA). Timed actions specify activities that are relevant for performance analysis. Immediate actions model logical events and the activities that are irrelevant from the performance viewpoint or much faster than others. Passive actions model activities waiting for the synchronization with timed or immediate ones, and express nondeterministic choice. The set of operators consist of *prefix*, functional *abstraction*, functional *relabeling*, *alternative* composition and *parallel* composition ones. Parallel composition includes *synchronization* on the set of action types. The syntax also includes *recursive* definitions given by means of constants. The semantics is interleaving and based on the labeled transition systems enriched with the information on action rates. For the exponentially timed kernel of the algebra (the sublanguage including only exponentially timed and passive actions), it is possible to construct CTMCs from the transition systems of the process terms to analyze the performance.

In dtsPBC, every activity is a pair consisting of the multiaction (not just an action, as in the classical SPAs) and its probability (not the rate, as in the classical SPAs) to be executed independently. dtsPBC has the sequence operation in contrast to the prefix one in the classical SPAs. One can combine arbitrary expressions with the sequence operator, i.e. it is more flexible than the prefix one, where the first argument should be a single activity. The choice operation in dtsPBC is similar to that in MTIPP, PEPA and to the alternative composition in EMPA, in the sense that the choice is probabilistic, but a discrete probability function is used in dtsPBC, unlike continuous ones in the classical calculi. Concurrency and synchronization in dtsPBC are different operations (this feature is inherited from PBC) in contrast to the classical SPAs, where parallel composition (combinator) has a synchronization capability. Relabeling in dtsPBC is analogous to that in EMPA, but it is additionally extended to conjugated actions. The restriction operation in dtsPBC differs from hiding in PEPA and functional abstraction in EMPA, where the hidden actions are labeled with a symbol of "silent" action τ. In dtsPBC, restriction by an action means that, for a given expression, any process behaviour containing the action or its conjugate is not allowed. The synchronization on an elementary action in dtsPBC collects all the pairs consisting of this elementary action and its conjugate which are contained in the multiactions from the synchronized activities. The operation produces new activities such that the first element of every resulting activity is the union of the multiactions from which all the mentioned pairs of conjugated actions are removed, and the second element is the product of the probabilities of the activities involved in the synchronization. This differs from the way synchronization is applied in the classical SPAs where it is accomplished over identical action names, and every resulting activity consists of the same action name and the rate calculated via some expression (including sums, minimums and products) on the rates of the initial activities, such as the apparent rate in PEPA. dtsPBC has no recursion operation or recursive definitions, but it includes the iteration operation to specify infinite looping behaviour with the explicit start and termination. dtsPBC has a discrete time semantics, and residence time in the states is geometrically distributed, unlike the classical SPAs with continuous time semantics and exponentially distributed activity delays. As a consequence, the semantics of dtsPBC is the step one in contrast to the interleaving semantics of the classical SPAs. The performance can be investigated based on the discrete time Markov chain (DTMC) extracted from the labeled probabilistic transition system associated with each expression of dtsPBC. In the classical SPAs, continuous time Markov chains (CTMCs) are used for performance evaluation. In (Bernardo *et al.* 1998), a denotational semantics of EMPA based on GSPNs has been defined, from which one can also extract the underlying SMCs and CTMCs (when both immediate and timed transitions are present) or discrete time Markov chains (DTMCs) (but when there are only immediate transitions). dtsPBC has a denotational semantics in terms of LDTSPNs from which the corresponding DTMCs can be derived as well.

9.2. *Continuous time and non-interleaving semantics*

Only a few non-interleaving SPAs were proposed among non-Markovian ones (Katoen and D'Argenio 2001). The semantics of all Markovian calculi is interleaving and their action delays have exponential distribution, which is the only continuous probability distribution with memoryless (Markovian) property. In (Brinksma *et al.* 1995), Generalized Stochastic Process Algebra (GSPA) was introduced. It has a true-concurrent denotational semantics in terms of generalized stochastic event structures (GSESs) with non-Markovian stochastic delays of events. In that paper, no operational semantics or performance evaluation methods for GSPA were presented. Later, in (Katoen *et al.* 1996), generalized semi-Markov processes (GSMPs) were extracted from GSESs to analyze performance. In (Priami 1996), Stochastic π-calculus (Sπ) with general continuous distributions of activity delays was defined. It has a proved operational semantics with transitions labeled by encodings of their deduction trees. No well-established underlying performance model for this version of Sπ was described. In (Bravetti *et al.* 1998), Generalized Semi-Markovian Process Algebra (GSMPA) was developed with ST-operational semantics and non-Markovian action delays. The performance analysis in GSMPA is accomplished via GSMPs.

Again, the first fundamental difference between dtsPBC and the calculi GSPA, Sπ and GSMPA is that dtsPBC is based on PBC, whereas GSPA is an extension of Process Algebra (PA) from (Brinksma *et al.* 1995), Sπ extends π-calculus (Milner *et al.* 1992) and GSMPA is an enrichment of EMPA. Therefore, both GSPA and GSMPA have prefixing, choice (alternative composition), parallel composition, renaming (relabeling) and hiding (abstraction) operations, but only GSMPA permits constants. Unlike dtsPBC, GSPA has neither iteration or recursion, GSMPA allows only recursive definitions, whereas Sπ additionally has operations to specify mobility. The second significant difference is that geometrically distributed delays are associated with process states in dtsPBC, unlike generally distributed delays assigned to events in GSPA or to activities in Sπ and GSMPA. As a consequence, dtsPBC has a discrete time operational semantics allowing for concurrent execution of activities in steps. GSPA has no operational semantics while Sπ and GSMPA have continuous time ones. In continuous time semantics, concurrency is simulated by interleaving, since simultaneous occurrence of any two events has zero probability according to the properties of continuous probability distributions. Therefore, interleaving transitions are often annotated with an additional information to keep concurrency data. The transition labels in the operational semantics of Sπ encode the action causality information and allow one to derive the enabling relations and the firing distributions of concurrent transitions from the transition sequences. At the same time, abstracting from stochastic delays leads to the classical early interleaving semantics of π-calculus. ST-operational semantics of GSMPA is based on decorated transition systems governed by transition rules with rather complex preconditions. There are two types of transitions: the choice (action beginning) and the termination (action ending) ones. The choice transitions are labeled by weights of single actions chosen for execution while the termination transitions have no labels. Only single actions can begin, but several actions can end in parallel. Thus, the choice transitions are the interleaving ones while the termination transitions are the step ones. As a result, the decorated interleaving / step transition systems are obtained. dtsPBC has an SPNs-based denotational semantics. In comparison with event structures, PNs are more expressive and visually tractable formalism capable of finitely specifying an infinite behaviour. Recursion in GSPA produces infinite GSESs while dtsPBC has iteration operation with a finite SPN semantics. An identification of infinite GSESs that can be finitely represented in GSPA was left for a future research.

9.3. *Discrete time*

In (van der Aalst *et al.* 2000), a class of compositional DTSPNs with generally distributed discrete time transition delays was proposed, called dts-nets. The denotational semantics of a stochastic extension (we call it stochastic ACP) of (a subset of) Algebra of Communicating Processes (ACP) (Bergstra and Klop 1985) can be constructed via dts-nets. There are two types of transitions in dts-nets: immediate (timeless) ones with zero delays and time ones, whose delays are random values having general discrete distributions. The top-down synthesis of dts-nets consists in the substitution of their transitions by blocks (dts-subnets) corresponding to the sequence, choice, parallelism and iteration operators. It was explained how to calculate the throughput time of dts-nets using the service time (defined as holding time or delay) of their transitions. For this, the notions of service distribution for the transitions and throughput distribution for the building blocks were defined. Since the throughput time of the parallelism block was calculated as the maximal service time for its two constituting transitions, the analogue of the step semantics approach was implemented. In (Markovski and de Vink 2008; Markovski and de Vink 2009), an SPA called Theory of Communicating Processes with discrete stochastic time (TCP^{dst}) was introduced. Its actions have a (deterministic) discrete real time delays (including zero time delays) or stochastic time delays. The algebra generalizes real-time processes to discrete stochastic time ones by applying real-time properties to stochastic time and imposing race condition to real time semantics. TCP^{dst} has an interleaving operational semantics in terms of stochastic transition systems. The performance is analyzed via discrete time probabilistic reward graphs which are essentially the reward transition systems with probabilistic states having finite number of outgoing probabilistic transitions and timed states having a single outgoing timed transition. The mentioned graphs can be transformed by unfolding or geometrization into discrete time Markov reward chains (DTMRCs) appropriate for transient or long-run (stationary) analysis.

The first difference between dtsPBC and the algebras stochastic ACP and TCP^{dst} is that dtsPBC is based on PBC, but stochastic ACP and TCP^{dst} are the extensions of ACP. Stochastic ACP has taken from ACP only sequence, choice, parallelism and iteration operations, whereas dtsPBC has additionally relabeling, restriction and synchronization ones, inherited from PBC. In TCP^{dst}, besides standard action prefixing, alternative, parallel composition, encapsulation (similar to restriction) and recursive variables, there are also timed delay prefixing, dependent delays scope and the maximal time progress operators, which are new both for ACP and dtsPBC. The second difference is that geometrically distributed delays are associated with process states in dtsPBC, unlike zero or generally distributed discrete time delays of actions in stochastic ACP and deterministic or generally distributed stochastic delays of actions in TCP^{dst}. Neither formal syntax nor operational semantics for stochastic ACP are defined and it is not explained how to derive Markov chains from the algebraic expressions or the corresponding dts-nets to analyze performance. It is not stated explicitly, which type of semantics (interleaving or step) is accommodated in stochastic ACP. In spite of the discrete time approach, operational semantics of TCP^{dst} is still interleaving, unlike that of dtsPBC. In addition, no denotational semantics was defined for TCP^{dst}.

Table 6 summarizes the SPAs comparison above and that from Section 1, by classifying the SPAs according to the concept of time and the type of operational semantics. The names of SPAs, whose denotational semantics is based on SPNs, are printed in bold font. The underlying stochastic process (if defined) is specified in parentheses near the name of the corresponding SPA.

Table 6. *Classification of stochastic process algebras*

Time	Interleaving semantics	Non-interleaving semantics
Continuous	MTIPP (CTMC), PEPA (CTMP), **EMPA** (SMC, CTMC), **sPBC** (CTMC), **gsPBC** (SMC)	GSPA (GSMP), Sπ, GSMPA (GSMP)
Discrete	TCP^{dst} (DTMRC)	**stochastic ACP**, **dtsPBC** (DTMC), **dtsiPBC** (SMC, DTMC)

10. Discussion

Let us now discuss which advantages has dtsPBC in comparison with the SPAs described in Section 9.

An important aspect is the analytical tractability of the underlying stochastic process, used for performance analysis within SPAs. The underlying CTMCs in MTIPP and PEPA, as well as SMCs in EMPA, are treated analytically, but these continuous time SPAs have just an interleaving semantics. GSPA, Sπ and GSMPA are the continuous time models, for which a non-interleaving semantics is constructed, but for the underlying GSMPs in GSPA and GSMPA, only simulation and numerical methods are applied, whereas no performance model for Sπ is defined. Stochastic ACP and TCP^{dst} are the discrete time models with the associated analytical methods for the throughput calculation in stochastic ACP or for the performance evaluation based on the underlying DTMRCs in TCP^{dst}, but both models have only an interleaving semantics. dtsPBC is a discrete time model with a non-interleaving semantics, where analytical methods are applied to the underlying DTMCs. Hence, if an interleaving model is appropriate as a framework for the analytical solution towards performance evaluation then one has a choice between the continuous time SPAs MTIPP, PEPA, EMPA and the discrete time ones stochastic ACP, TCP^{dst}. Otherwise, if one needs a non-interleaving model with the associated analytical methods for performance evaluation and the discrete time approach is feasible then dtsPBC is the right choice.

From the application viewpoint, one considers what kind of systems are more appropriate to be modeled and analyzed within SPAs. MTIPP and PEPA are well-suited for the interleaving continuous time systems such that the activity rates or the average sojourn time in the states are known in advance and exponential distribution approximates well the activity delay distributions, whereas EMPA can be used to model the mentioned systems with the activity delays of different duration order or the extended systems, in which purely probabilistic choices or urgent activities must be implemented. GSPA and GSMPA fit well for modeling the continuous time systems with a capability to keep the activity causality information, and with the known activity delay distributions, which cannot be approximated accurately by exponential distribution, while Sπ can additionally model mobility in such systems. TCP^{dst} is a good choice for interleaving discrete time systems with deterministic (fixed) and generalized stochastic delays, whereas stochastic ACP is capable to model non-interleaving systems as well, but it offers not enough performance analysis methods. dtsPBC is consistent for the step discrete time systems such that the independent execution probabilities of activities are known and geometrical distribution approximates well the state residence time distributions.

One can see that the stochastic process calculi proposed in the literature are based on interleaving, as a rule, and parallelism is simulated by synchronous or asynchronous execution. As their semantic domain, the interleaving formalism of transition systems is often used. Therefore, investigation of stochastic extensions for more expressive and powerful algebraic calculi is an important issue. The development of step or "true concurrency" (such that parallelism is considered as a causal independence) SPAs is an interesting and nontrivial problem, which has attracted special attention last years. Nevertheless, not so many formal stochastic models were defined whose underlying stochastic processes are DTMCs. As mentioned in (Fourneau 2010), such models are more difficult to analyze, since a lot of events can occur simultaneously in discrete time systems (the models have a step semantics) and the probability of a set of events can be not easily related to the probability of the single ones. As observed in (Horváth *et al.* 2012), even for stochastic models with generally distributed time delays, some restrictions on the concurrency degree were imposed to simplify their analysis techniques. In particular, the enabling restriction requires that no two generally distributed transitions are enabled in any reachable marking. Hence, their activity periods do not intersect and no two such transitions can fire simultaneously, this results in interleaving semantics of the model.

Stochastic models with discrete time and step semantics have the following important advantage in comparison with those having just interleaving semantics. The underlying Markov chains of parallel stochastic processes have the additional transitions corresponding to the simultaneous execution of concurrent (i.e. non-synchronized) activities. The transitions of that kind allow one to bypass a lot of intermediate states, which otherwise should be visited when interleaving semantics is accommodated. When step semantics is used, the intermediate states can also be visited with some probability (this is an advantage, since some alternative system's behaviour may start from these states), but this probability is not greater than the corresponding one in case of interleaving semantics. While in interleaving semantics, only the empty or singleton (multi)sets of activities can be executed, in step semantics, generally, the (multi)sets of activities with more than one element can be executed as well. Hence, in step semantics, there are more variants of execution from each state than in the interleaving case and the executions probabilities, whose sum should be equal to 1, are distributed among more possibilities. Therefore, the systems with parallel stochastic processes usually have smaller average run-through. In case the underlying Markov chains of the processes are ergodic, they will take less discrete time units to stabilize the behaviour, since their TPMs will be denser because of additional non-zero elements outside the main diagonal. Hence, both the first passage-time performance indices based on the transient probabilities and the steady-state performance indices based on the stationary probabilities can be computed quicker, resulting in faster quantitative analysis of the systems. On the other hand, step semantics, induced by simultaneous firing several transitions at each step, is natural for Petri nets and allows one to exploit full power of the model.

Thus, the main advantages of dtsPBC are the flexible multiaction labels and the set of powerful operations, as well as a step operational and a Petri net denotational semantics allowing for concurrent execution of activities (transitions), together with an ability for analytical performance evaluation.

11. Conclusion

In this paper, within the context of dtsPBC with iteration, we have defined the stochastic algebraic equivalences having natural net analogues on LDTSPNs. The diagram of interrelations for the algebraic equivalences has been constructed. We have explained how one can reduce transition systems and DTMCs as well as expressions and dts-boxes modulo the stochastic equivalences. We have investigated which of the equivalences we proposed guarantee identity of the stationary behaviour. We have proved that the weakest among the relations we have considered, step stochastic bisimulation equivalence, has this property. A case study of the dining philosophers system has been presented as an example of modeling, performance evaluation and performance preserving reduction in the framework of the calculus.

The advantage of our framework is twofold. First, one can specify in it concurrent composition and synchronization of (multi)actions, whereas this is not possible in classical Markov chains. Second, algebraic formulas represent processes in a more compact way than PNs and allow one to apply syntactic transformations and comparisons. Process algebras are compositional by definition and their operations naturally correspond to operators of programming languages. Hence, it is much easier to construct a complex model in the algebraic setting than in PNs. The complexity of PNs generated for practical models in the literature demonstrates that it is not straightforward to construct such PNs directly from the system specifications. dtsPBC is well suited for the discrete time applications, such as business processes, neural and transportation networks, computer and communication systems, whose discrete states change with a global time tick, as well as for those, in which the distributed architecture or the concurrency level should be preserved while modeling and analysis (remember that, in step semantics, we have additional transitions due to concurrent executions).

In the future, we plan to provide the stochastic equivalences with a logical characterization via probabilistic modal logics. Abstracting from the silent activities in the definitions of the equivalences, i.e. from the activities with empty multiaction part, is the next research direction. The main point here is that we should collect probabilities during such abstractions from an internal activity. Moreover, we plan to extend dtsPBC with recursion to enhance the specification power of the calculus.

Appendix A. Proofs

A.1. *Proof of Proposition 5.1*

Like it has been done for strong equivalence in Proposition 8.2.1 from (Hillston 1996), we shall prove the following fact about step stochastic bisimulation. Let us have for all $j \in \mathcal{J}$ it holds that $\mathcal{R}_j : G \underline{\leftrightarrow}_{ss} G'$ for some index set $\mathcal{J}$. Then the transitive closure of the union of all relations $\mathcal{R} = (\cup_{j\in\mathcal{J}}\mathcal{R}_j)^*$ is also an equivalence and $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$.

Since for all $j \in \mathcal{J}$ we have that $\mathcal{R}_j$ is an equivalence, by definition of $\mathcal{R}$, we get that $\mathcal{R}$ is also an equivalence.

Let $j \in \mathcal{J}$, then, by definition of $\mathcal{R}$, $(s_1, s_2) \in \mathcal{R}_j$ implies $(s_1, s_2) \in \mathcal{R}$. Hence, for all $\mathcal{H}_{jk} \in (DR(G) \cup DR(G'))/_{\mathcal{R}_j}$ there exists $\mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ such that $\mathcal{H}_{jk} \subseteq \mathcal{H}$. Moreover, there exists $\mathcal{J}'$ such that $\mathcal{H} = \cup_{k \in \mathcal{J}'} \mathcal{H}_{jk}$.

We denote $\mathcal{R}(n) = (\cup_{j \in \mathcal{J}} \mathcal{R}_j)^n$. Let $(s_1, s_2) \in \mathcal{R}$, then, by definition of $\mathcal{R}$, there exists $n > 0$ such that $(s_1, s_2) \in \mathcal{R}(n)$. We shall prove that $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$ by induction on n.

It is clear that for all $j \in \mathcal{J}$ the fact $\mathcal{R}_j : G \underline{\leftrightarrow}_{ss} G'$ implies that for all $j \in \mathcal{J}$ it holds that $([G]_\approx, [G']_\approx) \in \mathcal{R}_j$ and we have $([G]_\approx, [G']_\approx) \in \mathcal{R}$ by definition of $\mathcal{R}$.

It remains to prove that $(s_1, s_2) \in \mathcal{R}$ implies the following: for all $\mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ and for all $A \in I\!N_f^{\mathcal{L}}$ we have $PM_A(s_1, \mathcal{H}) = PM_A(s_2, \mathcal{H})$.

— $n = 1$

In this case, $(s_1, s_2) \in \mathcal{R}$ implies that there exists $j \in \mathcal{J}$ such that $(s_1, s_2) \in \mathcal{R}_j$. Since $\mathcal{R}_j : G \underline{\leftrightarrow}_{ss} G'$, we get for all $\mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ and for all $A \in I\!N_f^{\mathcal{L}}$ we have

$$PM_A(s_1, \mathcal{H}) = \sum_{k \in \mathcal{J}'} PM_A(s_1, \mathcal{H}_{jk}) = \sum_{k \in \mathcal{J}'} PM_A(s_2, \mathcal{H}_{jk}) = PM_A(s_2, \mathcal{H}).$$

— $n \to n + 1$

Suppose that for all $m \leq n$ the fact that $(s_1, s_2) \in \mathcal{R}(m)$ implies that for all $\mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ and for all $A \in I\!N_f^{\mathcal{L}}$ it holds that $PM_A(s_1, \mathcal{H}) = PM_A(s_2, \mathcal{H})$. Then $(s_1, s_2) \in \mathcal{R}(n+1)$ implies that there exists $j \in \mathcal{J}$ such that $(s_1, s_2) \in \mathcal{R}_j \circ \mathcal{R}(n)$, i.e. there exists $s_3 \in (DR(G) \cup DR(G'))$ such that $(s_1, s_3) \in \mathcal{R}_j$ and $(s_3, s_2) \in \mathcal{R}(n)$. Then, like for the case $n = 1$, we get $PM_A(s_1, \mathcal{H}) = PM_A(s_3, \mathcal{H})$. By the induction hypothesis, $PM_A(s_3, \mathcal{H}) = PM_A(s_2, \mathcal{H})$. Thus, for all $\mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ and for all $A \in I\!N_f^{\mathcal{L}}$ we have

$$PM_A(s_1, \mathcal{H}) = PM_A(s_3, \mathcal{H}) = PM_A(s_2, \mathcal{H}).$$

By definition, $\mathcal{R}_{ss}(G, G')$ is at least as large as the largest step stochastic bisimulation between G and G'. It follows from the proven above that $\mathcal{R}_{ss}(G, G')$ is an equivalence and $\mathcal{R}_{ss}(G, G') : G \underline{\leftrightarrow}_{ss} G'$, hence, it is the largest step stochastic bisimulation between G and G'. □

A.2. *Proof of Proposition 7.1*

It is sufficient to prove the statement of the proposition for transient PMFs only, since $\psi = \lim_{k \to \infty} \psi[k]$ and $\psi' = \lim_{k \to \infty} \psi'[k]$. We proceed by induction on k.

— $k = 0$

Note that the only nonzero values of the initial PMFs of $DTMC(G)$ and $DTMC(G')$ are $\psi[0]([G]_\approx)$ and $\psi[0]([G']_\approx)$. Let $\mathcal{H}_0$ be the equivalence class containing $[G]_\approx$ and $[G']_\approx$. Then $\sum_{s \in \mathcal{H}_0 \cap DR(G)} \psi[0](s) = \psi[0]([G]_\approx) = 1 = \psi'[0]([G']_\approx) = \sum_{s' \in \mathcal{H}_0 \cap DR(G')} \psi'[0](s')$.

As for other equivalence classes, for all $\mathcal{H} \in ((DR(G) \cup DR(G'))/_{\mathcal{R}}) \setminus \mathcal{H}_0$ we have $\sum_{s\in\mathcal{H}\cap DR(G)} \psi[0](s) = 0 = \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'[0](s')$.

— $k \to k+1$

Let $\mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ and $s_1, s_2 \in \mathcal{H}$. We have for all $\widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ and for all $A \in I\!N_f^{\mathcal{L}}$ it holds that $s_1 \xrightarrow{A}_{\mathcal{P}} \widetilde{\mathcal{H}}$ iff $s_2 \xrightarrow{A}_{\mathcal{P}} \widetilde{\mathcal{H}}$.
Therefore, $PM(s_1, \widetilde{\mathcal{H}}) = \sum_{\{\Gamma \mid \exists \tilde{s}_1 \in \widetilde{\mathcal{H}}\ s_1 \xrightarrow{\Gamma} \tilde{s}_1\}} PT(\Gamma, s_1) =$
$\sum_{A\in I\!N_f^{\mathcal{L}}} \sum_{\{\Gamma \mid \exists \tilde{s}_1 \in \widetilde{\mathcal{H}}\ s_1 \xrightarrow{\Gamma} \tilde{s}_1,\ \mathcal{L}(\Gamma)=A\}} PT(\Gamma, s_1) = \sum_{A\in I\!N_f^{\mathcal{L}}} PM_A(s_1, \widetilde{\mathcal{H}}) =$
$\sum_{A\in I\!N_f^{\mathcal{L}}} PM_A(s_2, \widetilde{\mathcal{H}}) = \sum_{A\in I\!N_f^{\mathcal{L}}} \sum_{\{\Gamma \mid \exists \tilde{s}_2 \in \widetilde{\mathcal{H}}\ s_2 \xrightarrow{\Gamma} \tilde{s}_2,\ \mathcal{L}(\Gamma)=A\}} PT(\Gamma, s_2) =$
$\sum_{\{\Gamma \mid \exists \tilde{s}_2 \in \widetilde{\mathcal{H}}\ s_2 \xrightarrow{\Gamma} \tilde{s}_2\}} PT(\Gamma, s_2) = PM(s_2, \widetilde{\mathcal{H}})$. Since we have the previous equality for all $s_1, s_2 \in \mathcal{H}$, we can denote $PM(\mathcal{H}, \widetilde{\mathcal{H}}) = PM(s_1, \widetilde{\mathcal{H}}) = PM(s_2, \widetilde{\mathcal{H}})$. Note that transitions from the states of $DR(G)$ always lead to those from the same set, hence, $\forall s \in DR(G)\ PM(s, \widetilde{\mathcal{H}}) = PM(s, \widetilde{\mathcal{H}} \cap DR(G))$. The same is true for $DR(G')$.
By induction hypothesis, $\sum_{s\in\mathcal{H}\cap DR(G)} \psi[k](s) = \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'[k](s')$. Further,
$\sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} \psi[k+1](\tilde{s}) = \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} \sum_{s\in DR(G)} \psi[k](s) PM(s, \tilde{s}) =$
$\sum_{s\in DR(G)} \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} \psi[k](s) PM(s, \tilde{s}) =$
$\sum_{s\in DR(G)} \psi[k](s) \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} PM(s, \tilde{s}) =$
$\sum_{\mathcal{H}} \sum_{s\in\mathcal{H}\cap DR(G)} \psi[k](s) \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} PM(s, \tilde{s}) =$
$\sum_{\mathcal{H}} \sum_{s\in\mathcal{H}\cap DR(G)} \psi[k](s) \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} \sum_{\{\Gamma \mid s \xrightarrow{\Gamma} \tilde{s}\}} PT(\Gamma, s) =$
$\sum_{\mathcal{H}} \sum_{s\in\mathcal{H}\cap DR(G)} \psi[k](s) \sum_{\{\Gamma \mid \exists \tilde{s} \in \widetilde{\mathcal{H}}\cap DR(G)\ s \xrightarrow{\Gamma} \tilde{s}\}} PT(\Gamma, s) =$
$\sum_{\mathcal{H}} \sum_{s\in\mathcal{H}\cap DR(G)} \psi[k](s) PM(s, \widetilde{\mathcal{H}}) = \sum_{\mathcal{H}} \sum_{s\in\mathcal{H}\cap DR(G)} \psi[k](s) PM(\mathcal{H}, \widetilde{\mathcal{H}}) =$
$\sum_{\mathcal{H}} PM(\mathcal{H}, \widetilde{\mathcal{H}}) \sum_{s\in\mathcal{H}\cap DR(G)} \psi[k](s) = \sum_{\mathcal{H}} PM(\mathcal{H}, \widetilde{\mathcal{H}}) \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'[k](s') =$
$\sum_{\mathcal{H}} \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'[k](s') PM(\mathcal{H}, \widetilde{\mathcal{H}}) = \sum_{\mathcal{H}} \sum_{s'\in\mathcal{H}'\cap DR(G')} \psi'[k](s') PM(s', \widetilde{\mathcal{H}}) =$
$\sum_{\mathcal{H}} \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'[k](s') \sum_{\{\Gamma \mid \exists \tilde{s}' \in \widetilde{\mathcal{H}}\cap DR(G')\ s' \xrightarrow{\Gamma} \tilde{s}'\}} PT(\Gamma, s') =$
$\sum_{\mathcal{H}} \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'[k](s') \sum_{\tilde{s}'\in\widetilde{\mathcal{H}}\cap DR(G')} \sum_{\{\Gamma \mid \exists \tilde{s}'\ s' \xrightarrow{\Gamma} \tilde{s}'\}} PT(\Gamma, s') =$
$\sum_{\mathcal{H}} \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'[k](s') \sum_{\tilde{s}'\in\widetilde{\mathcal{H}}\cap DR(G')} PM(s', \tilde{s}') =$
$\sum_{s'\in DR(G')} \psi'[k](s') \sum_{\tilde{s}'\in\widetilde{\mathcal{H}}\cap DR(G')} PM(s', \tilde{s}') =$
$\sum_{s'\in DR(G')} \sum_{\tilde{s}'\in\widetilde{\mathcal{H}}\cap DR(G')} \psi'[k](s') PM(s', \tilde{s}') =$
$\sum_{\tilde{s}'\in\widetilde{\mathcal{H}}\cap DR(G')} \sum_{s'\in DR(G')} \psi'[k](s') PM(s', \tilde{s}') = \sum_{\tilde{s}'\in\widetilde{\mathcal{H}}\cap DR(G')} \psi'[k+1](\tilde{s}')$. □

A.3. *Proof of Theorem 7.1*

Let $\mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ and $s, \bar{s} \in \mathcal{H}$. We have for all $\widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ and for all $A \in I\!N_f^{\mathcal{L}}$ it holds that $s \xrightarrow{A}_{\mathcal{P}} \widetilde{\mathcal{H}}$ iff $\bar{s} \xrightarrow{A}_{\mathcal{P}} \widetilde{\mathcal{H}}$. The previous statement is valid for all $s, \bar{s} \in \mathcal{H}$, hence, we can rewrite it as $\mathcal{H} \xrightarrow{A}_{\mathcal{P}} \widetilde{\mathcal{H}}$ and denote $PM_A(\mathcal{H}, \widetilde{\mathcal{H}}) = PM_A(s, \widetilde{\mathcal{H}}) = PM_A(\bar{s}, \widetilde{\mathcal{H}})$. Note that transitions from the states of $DR(G)$ always lead to those from the same set, hence, for all $s \in DR(G)$ we have $PM_A(s, \widetilde{\mathcal{H}}) = PM_A(s, \widetilde{\mathcal{H}} \cap DR(G))$. The same is true for $DR(G')$.

Let $\Sigma = A_1 \cdots A_n$ be a derived step trace of G and G'. Then there exist $\mathcal{H}_0, \ldots, \mathcal{H}_n \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ such that $\mathcal{H}_0 \xrightarrow{A_1}_{\mathcal{P}_1} \mathcal{H}_1 \xrightarrow{A_2}_{\mathcal{P}_2} \cdots \xrightarrow{A_n}_{\mathcal{P}_n} \mathcal{H}_n$. Now we prove that the sum of probabilities of all the paths starting in every $s_0 \in \mathcal{H}_0$ and going through the states from $\mathcal{H}_1, \ldots, \mathcal{H}_n$ is equal to the product of $\mathcal{P}_1, \ldots, \mathcal{P}_n$:

$$\sum_{\{\Gamma_1,\ldots,\Gamma_n \mid s_0 \xrightarrow{\Gamma_1} \cdots \xrightarrow{\Gamma_n} s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\leq i\leq n)\}} \prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}) = \prod_{i=1}^{n} PM_{A_i}(\mathcal{H}_{i-1}, \mathcal{H}_i).$$

We prove this equality by induction on the derived step trace length n.

— $n = 1$

$\sum_{\{\Gamma_1 \mid s_0 \xrightarrow{\Gamma_1} s_1,\ \mathcal{L}(\Gamma_1)=A_1,\ s_1\in\mathcal{H}_1\}} PT(\Gamma_1, s_0) = PM_{A_1}(s_0, \mathcal{H}_1) = PM_{A_1}(\mathcal{H}_0, \mathcal{H}_1).$

— $n \to n+1$

$\sum_{\{\Gamma_1,\ldots,\Gamma_n,\Gamma_{n+1} \mid s_0 \xrightarrow{\Gamma_1} \cdots \xrightarrow{\Gamma_n} s_n \xrightarrow{\Gamma_{n+1}} s_{n+1},\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\leq i\leq n+1)\}} \prod_{i=1}^{n+1} PT(\Gamma_i, s_{i-1}) =$
$\sum_{\{\Gamma_1,\ldots,\Gamma_n \mid s_0 \xrightarrow{\Gamma_1} \cdots \xrightarrow{\Gamma_n} s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\leq i\leq n)\}}$
$\sum_{\{\Gamma_{n+1} \mid s_n \xrightarrow{\Gamma_{n+1}} s_{n+1},\ \mathcal{L}(\Gamma_{n+1})=A_{n+1},\ s_n\in\mathcal{H}_n,\ s_{n+1}\in\mathcal{H}_{n+1}\}} \prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}) PT(\Gamma_{n+1}, s_n) =$
$\sum_{\{\Gamma_1,\ldots,\Gamma_n \mid s_0 \xrightarrow{\Gamma_1} \cdots \xrightarrow{\Gamma_n} s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\leq i\leq n)\}} \Big[\prod_{i=1}^{n} PT(\Gamma_i, s_{i-1})$
$\sum_{\{\Gamma_{n+1} \mid s_n \xrightarrow{\Gamma_{n+1}} s_{n+1},\ \mathcal{L}(\Gamma_{n+1})=A_{n+1},\ s_n\in\mathcal{H}_n,\ s_{n+1}\in\mathcal{H}_{n+1}\}} PT(\Gamma_{n+1}, s_n)\Big] =$
$\sum_{\{\Gamma_1,\ldots,\Gamma_n \mid s_0 \xrightarrow{\Gamma_1} \cdots \xrightarrow{\Gamma_n} s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\leq i\leq n)\}}$
$\prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}) PM_{A_{n+1}}(s_n, \mathcal{H}_{n+1}) =$
$\sum_{\{\Gamma_1,\ldots,\Gamma_n \mid s_0 \xrightarrow{\Gamma_1} \cdots \xrightarrow{\Gamma_n} s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\leq i\leq n)\}}$
$\prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}) PM_{A_{n+1}}(\mathcal{H}_n, \mathcal{H}_{n+1}) =$
$PM_{A_{n+1}}(\mathcal{H}_n, \mathcal{H}_{n+1}) \sum_{\{\Gamma_1,\ldots,\Gamma_n \mid s_0 \xrightarrow{\Gamma_1} \cdots \xrightarrow{\Gamma_n} s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\leq i\leq n)\}}$
$\prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}) =$
$PM_{A_{n+1}}(\mathcal{H}_n, \mathcal{H}_{n+1}) \prod_{i=1}^{n} PM_{A_i}(\mathcal{H}_{i-1}, \mathcal{H}_i) = \prod_{i=1}^{n+1} PM_{A_i}(\mathcal{H}_{i-1}, \mathcal{H}_i).$

Let $s_0, \bar{s}_0 \in \mathcal{H}_0$. We have $PT(A_1 \cdots A_n, s_0) =$
$\sum_{\{\Gamma_1,\ldots,\Gamma_n \mid s_0 \xrightarrow{\Gamma_1} \cdots \xrightarrow{\Gamma_n} s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ (1\leq i\leq n)\}} \prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}) =$
$\sum_{\mathcal{H}_1,\ldots,\mathcal{H}_n} \sum_{\{\Gamma_1,\ldots,\Gamma_n \mid s_0 \xrightarrow{\Gamma_1} \cdots \xrightarrow{\Gamma_n} s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\leq i\leq n)\}} \prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}) =$
$\sum_{\mathcal{H}_1,\ldots,\mathcal{H}_n} \prod_{i=1}^{n} PM_{A_i}(\mathcal{H}_{i-1}, \mathcal{H}_i) =$
$\sum_{\mathcal{H}_1,\ldots,\mathcal{H}_n} \sum_{\{\overline{\Gamma}_1,\ldots,\overline{\Gamma}_n \mid \bar{s}_0 \xrightarrow{\overline{\Gamma}_1} \cdots \xrightarrow{\overline{\Gamma}_n} \bar{s}_n,\ \mathcal{L}(\overline{\Gamma}_i)=A_i,\ \bar{s}_i\in\mathcal{H}_i\ (1\leq i\leq n)\}} \prod_{i=1}^{n} PT(\overline{\Gamma}_i, \bar{s}_{i-1}) =$
$\sum_{\{\overline{\Gamma}_1,\ldots,\overline{\Gamma}_n \mid \bar{s}_0 \xrightarrow{\overline{\Gamma}_1} \cdots \xrightarrow{\overline{\Gamma}_n} \bar{s}_n,\ \mathcal{L}(\overline{\Gamma}_i)=A_i,\ (1\leq i\leq n)\}} \prod_{i=1}^{n} PT(\overline{\Gamma}_i, \bar{s}_{i-1}) = PT(A_1 \cdots A_n, \bar{s}_0).$

Since we have the previous equality for all $s_0, \bar{s}_0 \in \mathcal{H}_0$, we can denote
$PT(A_1 \cdots A_n, \mathcal{H}_0) = PT(A_1 \cdots A_n, s_0) = PT(A_1 \cdots A_n, \bar{s}_0)$.

By Proposition 7.1, $\sum_{s\in\mathcal{H}\cap DR(G)} \psi(s) = \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'(s')$. Now we can complete the proof: $\sum_{s\in\mathcal{H}\cap DR(G)} \psi(s) PT(\Sigma, s) = \sum_{s\in\mathcal{H}\cap DR(G)} \psi(s) PT(\Sigma, \mathcal{H}) =$
$PT(\Sigma, \mathcal{H}) \sum_{s\in\mathcal{H}\cap DR(G)} \psi(s) = PT(\Sigma, \mathcal{H}) \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'(s') =$
$\sum_{s'\in\mathcal{H}\cap DR(G')} \psi'(s') PT(\Sigma, \mathcal{H}) = \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'(s') PT(\Sigma, s').$ □

Acknowledgements I would like to thank Eike Best for the encouraging discussions and valuable advices related to the subject of this paper. I am grateful to Peter Buchholz and Falko Bause for interest to my work and the qualified consideration. Many thanks to anonymous referees for their helpful suggestions on improvement of the paper.

References

van der Aalst, W.M.P., van Hee, K.M. and Reijers, H.A. (2000): Analysis of discrete-time stochastic Petri nets. *Statistica Neerlandica* **54**(2), 237–255, http://tmitwww.tm.tue.nl/staff/hreijers/H.A. Reijers Bestanden/Statistica.pdf.

Autant, C. and Schnoebelen, Ph. (1992): Place bisimulations in Petri nets. *Proc. of* 13^{th} *ICATPN 1992*, *Lect. Notes Comp. Sci.* **616**, 45–61.

Baarir, S., Beccuti, M., Dutheillet, C., Franceschinis, G. and Haddad, S. (2011): Lumping partially symmetric stochastic models. *Performance Evaluation* **68**, 21–44.

Bergstra, J.A. and Klop, J.W. (1985): Algebra of communicating processes with abstraction. *Theor. Comput. Sci.* **37**, 77–121.

Bernardo, M. (2007): A survey of Markovian behavioral equivalences. *Proc. of* 7^{th} *SFM 2007*, *Lect. Notes Comp. Sci.* **4486**, 180–219, http://www.sti.uniurb.it/bernardo/documents/sfm07pe.pdf.

Bernardo, M., Donatiello, L. and Gorrieri, R. (1998): A formal approach to the integration of performance aspects in the modeling and analysis of concurrent systems. *Information and Computation* **144**(2), 83–154, http://www.sti.uniurb.it/bernardo/documents/ic144.pdf.

Bernardo, M. and Gorrieri, R. (1998): A tutorial on EMPA: a theory of concurrent processes with nondeterminism, priorities, probabilities and time. *Theor. Comput. Sci.* **202**, 1–54.

Best, E., Devillers, R. and Hall, J.G. (1992): The box calculus: a new causal algebra with multi-label communication. *Advances in Petri Nets 1992*, *Lect. Notes Comp. Sci.* **609**, 21–69.

Best, E., Devillers, R. and Koutny, M. (2001): *Petri net algebra. EATCS Monographs on Theor. Comput. Sci.*, 378 pages, Springer Verlag.

Best, E. and Koutny, M. (1995): A refined view of the box algebra. *Proc. of* 16^{th} *ICATPN 1995*, *Lect. Notes Comp. Sci.* **935**, 1–20.

Bravetti, M., Bernardo, M. and Gorrieri, R. (2008): Towards performance evaluation with general distributions in process algebras. *Proc. of* 9^{th} *CONCUR 1998*, *Lect. Notes Comp. Sci.* **1466**, 405–422, http://www.cs.unibo.it/~bravetti/papers/concur98.ps.

Brinksma, E. and Hermanns, H. (2001): Process algebra and Markov chains. *Proc. of* 1^{st} *EEF/Euro Summer School of Trends in Comp. Sci. 2000*, *Lect. Notes Comp. Sci.* **2090**, 183–231.

Brinksma, E., Katoen, J.-P., Langerak, R. and Latella, D. (1995): A stochastic causality-based process algebra. *The Computer Journal* **38**(7), 552–565, http://eprints.eemcs.utwente.nl/6387/01/552.pdf.

Buchholz, P. (1995): A notion of equivalence for stochastic Petri nets. *Proc. of* 16^{th} *ICATPN 1995*, *Lect. Notes Comp. Sci.* **935**, 161–180.

Buchholz, P. (1998): Iterative decomposition and aggregation of labeled GSPNs. *Proc. of* 19^{th} *ICATPN 1998*, *Lect. Notes Comp. Sci.* **1420**, 226–245.

Buchholz, P. and Tarasyuk, I.V. (2001): Net and algebraic approaches to probabilistic modeling. *Joint Novosibirsk Computing Center and Institute of Informatics Systems Bulletin, Series Computer Science* **15**, 31–64, Novosibirsk, http://itar.iis.nsk.su/files/itar/pages/spnpancc.pdf.

Derisavi, S., Hermanns, H. and Sanders, W.H. (2003): Optimal state-space lumping of Markov chains. *Information Processing Letters* **87**(6), 309–315.

Fourneau, J.M. (2010): Collaboration of discrete-time Markov chains: Tensor and product form. *Performance Evaluation* **67**, 779–796.

van Glabbeek, R.J., Smolka, S.A. and Steffen, B. (1995): Reactive, generative, and stratified models of probabilistic processes. *Information and Computation* **121**(1), 59–80, http://boole.stanford.edu/pub/prob.ps.gz.

Hermanns, H. and Rettelbach, M. (1994): Syntax, semantics, equivalences and axioms for MTIPP. *Proc. of* 2^{nd} *International Workshop on Process Algebras and Performance Modelling (PAPM) 1994* (Herzog U., Rettelbach M., eds.), *Arbeitsberichte des IMMD* **27**, 71–88, University of Erlangen, Germany.

Hillston, J. (1994): *The nature of synchronisation. Proc. of* 2^{nd} *International Workshop on Process Algebras and Performance Modelling (PAPM) 1994* (Herzog U., Rettelbach M., eds.), *Arbeitsberichte des IMMD* **27**, 51–70, University of Erlangen, Germany, http://www.dcs.ed.ac.uk/pepa/synchronisation.pdf.

Hillston, J. (1996): *A compositional approach to performance modelling.* Cambridge University Press, UK.

Horváth, A., Paolieri, M., Ridi, L. and Vicario, E. (2012): Transient analysis of non-Markovian models using stochastic state classes. *Performance Evaluation* **69**(7–8), 315–335.

Jou, C.-C. and Smolka, S.A. (1990): Equivalences, congruences and complete axiomatizations for probabilistic processes. *Proc. of* 1^{st} *CONCUR 1990*, *Lect. Notes Comp. Sci.* **458**, 367–383.

Katoen, J.-P., Brinksma, E., Latella, D. and Langerak, R. (1996): Stochastic simulation of event structures. *Proc. of* 4^{th} *International Workshop on Process Algebra and Performance Modelling (PAPM) 1996* (M. Ribaudo, ed.), 21–40, CLUT Press, Torino, Italy, http://eprints.eemcs.utwente.nl/6487/01/263_KLLB96b.pdf.

Katoen, J.-P. and D'Argenio, P.R. (2001): General distributions in process algebra. *Proc. of* 1^{st} *EEF/Euro Summer School of Trends in Comp. Sci. 2000*, *Lect. Notes Comp. Sci.* **2090**, 375–429.

Katoen, J.-P., Zapreev, I.S., Hahn, E.M., Hermanns, H. and Jansen, D.N. (2011): The ins and outs of the probabilistic model checker MRMC. *Performance Evaluation* **68**, 90–104.

Koutny, M. (2000): A compositional model of time Petri nets. *Proc. of* 21^{st} *ICATPN 2000*, *Lect. Notes Comp. Sci.* **1825**, 303–322.

Kulkarni, V.G. (2009): Modeling and analysis of stochastic systems. *Texts in Statistical Science*, 563 pages, Chapman and Hall / CRC Press.

Kwiatkowska, M.Z., Norman, G.J. and Parker, D. (2006): Symmetry reduction for probabilistic model checking. *Proc. of ICIC 2006*, *Lect. Notes Comp. Sci.* **4144**, 234–248.

Larsen, K.G. and Skou, A. (1991): Bisimulation through probabilistic testing. *Information and Computation* **94**(1), 1–28.

Macià, H., Valero, V., Cazorla, D. and Cuartero, F. (2004): Introducing the iteration in sPBC. *Proc. of* 24^{th} *FORTE 2004*, *Lect. Notes Comp. Sci.* **3235**, 292–308.

Macià, H., Valero, V., Cuartero, F. and Ruiz, M.C. (2008): sPBC: a Markovian extension of Petri box calculus with immediate multiactions. *Fundamenta Informaticae* **87**(3–4), 367–406, IOS Press, Amsterdam, The Netherlands.

Macià, H., Valero, V. and de-Frutos, D. (2001): sPBC: a Markovian extension of finite Petri box calculus. *Proc. of* 9^{th} *IEEE Int. Workshop on Petri Nets and Performance Models (PNPM) 2001*, 207–216, Aachen, Germany, IEEE Computer Society Press.

Markovski, J. and de Vink, E.P. (2008): Extending timed process algebra with discrete stochastic time. *Proc. of* 12^{th} *AMAST 2008*, *Lect. Notes Comp. Sci.* **5140**, 268–283.

Markovski, J. and de Vink, E.P. (2009): Performance evaluation of distributed systems based on a discrete real- and stochastic-time process algebra. *Fundamenta Informaticae* **95**(1), 157–186, IOS Press, Amsterdam, The Netherlands.

Marroquín, O. and de-Frutos, D. (2001): Extending the Petri box calculus with time. *Proc. of* 22^{nd} *ICATPN 2001*, *Lect. Notes Comp. Sci.* **2075**, 303–322.

Marsan, M.A. (1990): Stochastic Petri nets: an elementary introduction. *Advances in Petri Nets 1989*, *Lect. Notes Comp. Sci.* **424**, 1–29.

Merlin, P. and Farber, D.J. (1976): Recoverability of communication protocols: implications of a theoretical study. *IEEE Transactions on Communications* **24**(9), 1036–1043.

Milner, R.A.J. (1989): *Communication and concurrency.* Prentice-Hall, 260 pages, Upper Saddle River, NJ, USA.

Milner, R.A.J., Parrow, J. and Walker, D. (1992): A calculus of mobile processes (I and II). *Information and Computation* **100**(1), 1–77.

Molloy, M.K. (1981): On the integration of the throughput and delay measures in distributed processing models. *Ph. D. thesis*, 108 pages, University of California, Los Angeles, USA.

Molloy, M.K. (1985): Discrete time stochastic Petri nets. *IEEE Transactions on Software Engineering* **11**(4), 417–423.

Niaouris, A. (2005): An algebra of Petri nets with arc-based time restrictions. *Proc. of* 1^{st} *ICTAC 2004*, *Lect. Notes Comp. Sci.* **3407**, 447–462.

Paige, R. and Tarjan, R.E. (1987): Three partition refinement algorithms. *SIAM J. Comput.* **16**(6), 973–989.

Peterson, J.L. (1981): *Petri net theory and modeling of systems.* Prentice-Hall.

Priami, C. (1996): Stochastic π-calculus with general distributions. *Proc. of* 4^{th} *International Workshop on Process Algebra and Performance Modelling (PAPM) 1996* (M. Ribaudo, ed.), CLUT Press, Torino, Italy, 41–57.

Ramchandani, C. (1973): Perfomance evaluation of asynchronous concurrent systems by timed Petri nets. *Ph. D. thesis*, Massachusetts Institute of Technology, Cambridge, USA.

Tarasyuk, I.V. (2005): Discrete time stochastic Petri box calculus. *Berichte aus dem Department für Informatik* **3/05**, 25 pages, Carl von Ossietzky Universität Oldenburg, Germany, http://itar.iis.nsk.su/files/itar/pages/dtspbcib_cov.pdf.

Tarasyuk, I.V. (2006): Iteration in discrete time stochastic Petri box calculus. *Bulletin of the Novosibirsk Computing Center, Series Computer Science, IIS Special Issue* **24**, 129–148, NCC Publisher, Novosibirsk, http://itar.iis.nsk.su/files/itar/pages/dtsitncc.pdf.

Tarasyuk, I.V. (2007): Stochastic Petri box calculus with discrete time. *Fundamenta Informaticae* **76**(1–2), 189–218, IOS Press, Amsterdam, The Netherlands.

Tarasyuk, I.V. (2008): Investigating equivalence relations in dtsPBC. *Berichte aus dem Department für Informatik* **5/08**, 57 pages, Carl von Ossietzky Universität Oldenburg, Germany, http://itar.iis.nsk.su/files/itar/pages/dtspbcit_cov.pdf.

Tarasyuk, I.V., Macià, H. and Valero, V. (2010): Discrete time stochastic Petri box calculus with immediate multiactions. *Technical Report* **DIAB-10-03-1**, 25 pages, Department of Computer Systems, High School of Computer Science Engineering, University of Castilla - La Mancha, Albacete, Spain, http://itar.iis.nsk.su/files/itar/pages/dtsipbc.pdf.

Wimmer, R., Derisavi, S. and Hermanns, H. (2010): Symbolic partition refinement with automatic balancing of time and space. *Performance Evaluation* **67**, 816–836.