RESEARCH ARTICLE

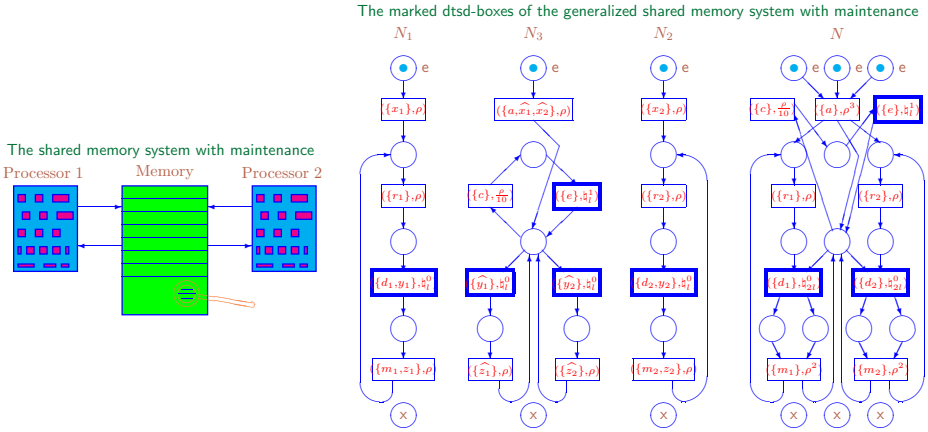
Performance analysis of the shared memory system in stochastic process algebra dtsdPBC

I. V. Tarasyuk[a]

[a]Laboratory for Theory of Concurrent Processes, A.P. Ershov Institute of Informatics Systems, Siberian Branch of the Russian Academy of Sciences, Acad. Lavrentiev pr. 6, 630090 Novosibirsk, Russian Federation

ARTICLE HISTORY
Compiled April 23, 2026

ABSTRACT
Discrete time stochastic and deterministic Petri box calculus (dtsdPBC) is a parallel process algebra with stochastic and deterministic delays. To evaluate performance in dtsdPBC, semi-Markov chains (SMCs) and (reduced) discrete time Markov chains (DTMCs/RDTMCs) are analyzed. Stochastic bisimulation equivalence is used for quotienting the transition systems, SMCs and DTMCs/RDTMCs of the process expressions while preserving stationary behaviour and residence time. Our example of generalized shared memory system with maintenance demonstrates modeling, performance analysis and reduction by quotienting. The generalized system takes the probabilities and weights from the standard system's specification as variables, adjusted for performance optimization.



KEYWORDS
Petri box calculus; stochastic and deterministic delays; Markov chain; performance analysis; stochastic bisimulation quotient; shared memory system

1. Introduction

Process calculi, like CSP [1], ACP [2] and CCS [3] are well-known formal models for specification of computing systems and analysis of their behaviour. In such process

CONTACT I. V. Tarasyuk. Email: itar@iis.nsk.su

algebras (PAs), formulas describe processes, and verification of the functionality properties of their behavior is accomplished at a syntactic level via equivalences, axioms and inference rules. In order to represent stochastic timing and analyze the performance properties, stochastic extensions of PAs were proposed, like MTIPP [4], PEPA [5] and EMPA [6]. Such stochastic process algebras (SPAs) specify actions which can occur (qualitative features) and associate with the actions the distribution parameters of their random delays (quantitative characteristics).

1.1. Petri box calculus (PBC)

Petri box calculus (PBC) [7–9] is a flexible and expressive process algebra developed as a tool for specification of the Petri nets (PNs) structure and their interrelations. Its goal was also to propose a compositional semantics for high level constructs of concurrent programming languages in terms of elementary PNs. Formulas of PBC are combined from multisets of elementary actions and their conjugates, called multiactions (*basic formulas*). The empty multiset of actions is interpreted as the silent multiaction specifying an invisible activity. The operational semantics of PBC is of step type, since its SOS rules have transitions with (multi)sets of activities, corresponding to simultaneous executions of activities (steps). A denotational semantics of PBC was proposed via a subclass of PNs with an interface and considered up to isomorphism, called Petri boxes. The extensions of PBC with a deterministic, a nondeterministic or a stochastic model of time exist.

1.2. Time extensions of PBC

A time extension of PBC with a nondeterministic time model, called time Petri box calculus (tPBC), was proposed in [10]. In tPBC, timing information is added by associating time intervals with instantaneous *actions*. tPBC has a step time operational semantics in terms of labeled transition systems. Its denotational semantics was defined in terms of a subclass of labeled time Petri nets (LtPNs), based on tPNs [11] and called time Petri boxes (ct-boxes).

Another time enrichment of PBC, called Timed Petri box calculus (TPBC), was defined in [12,13], it accommodates a deterministic model of time. In contrast to tPBC, multiactions of TPBC are not instantaneous, but have time durations. TPBC has a step timed operational semantics in terms of labeled transition systems. The denotational semantics of TPBC was defined in terms of a subclass of labeled Timed Petri nets (LTPNs), based on TPNs [14] and called Timed Petri boxes (T-boxes).

The third time extension of PBC, called arc time Petri box calculus (atPBC), was constructed in [15,16], and it implements a nondeterministic time. In atPBC, multiactions are associated with time delay intervals. atPBC possesses a step time operational semantics in terms of labeled transition systems. Its denotational semantics was defined on a subclass of labeled arc time Petri nets (atPNs), based of those from [17,18], where time restrictions are associated with the arcs, called arc time Petri boxes (at-boxes). tPBC, TPBC and atPBC, all adapt discrete time, but TPBC has no immediate (multi)actions (those with zero delays).

1.3. Stochastic extensions of PBC

A stochastic extension of PBC, called stochastic Petri box calculus (sPBC), was proposed in [19–21]. In sPBC, multiactions have stochastic delays that follow (negative) exponential distribution. Each multiaction is equipped with a rate that is a parameter of the corresponding exponential distribution. The (instantaneous) execution of a stochastic multiaction is possible only after the corresponding stochastic time delay. The calculus has an interleaving operational semantics defined via transition systems labeled with multiactions and their rates. Its denotational semantics was defined in terms of a subclass of labeled continuous time stochastic PNs, based on CTSPNs [22,23] and called stochastic Petri boxes (s-boxes).

sPBC was enriched with immediate multiactions having zero delay in [24,25]. We call such an extension generalized sPBC (gsPBC). An interleaving operational semantics of gsPBC was constructed via transition systems labeled with stochastic or immediate multiactions together with their rates or probabilities. A denotational semantics of gsPBC was defined via a subclass of labeled generalized stochastic PNs, based on GSPNs [22,23,26] and called generalized stochastic Petri boxes (gs-boxes).

In [27–30], we presented a discrete time stochastic extension dtsPBC of the algebra PBC. In dtsPBC, the residence time in the process states is geometrically distributed. A step operational semantics of dtsPBC was constructed via labeled probabilistic transition systems. Its denotational semantics was defined in terms of a subclass of labeled discrete time stochastic PNs (LDTSPNs), based on DTSPNs [31,32] and called discrete time stochastic Petri boxes (dts-boxes).

In [33–37], a calculus dtsiPBC was proposed as an extension with immediate multiactions of dtsPBC. Immediate multiactions increase the specification capability: they can model logical conditions, probabilistic branching, instantaneous probabilistic choices and activities whose durations are negligible in comparison with those of others. They are also used to specify urgent activities and the ones that are not relevant for performance evaluation. The step operational semantics of dtsiPBC was constructed with the use of labeled probabilistic transition systems. Its denotational semantics was defined in terms of a subclass of labeled discrete time stochastic and immediate PNs (LDTSIPNs), called dtsi-boxes.

In [38–42], we defined dtsdPBC, an extension of dtsiPBC with deterministic multiactions. In dtsdPBC, besides the probabilities from the real-valued interval $(0; 1)$, applied to calculate discrete time delays of stochastic multiactions, also non-negative integers are used to specify fixed delays of deterministic multiactions (including zero delay, which is the case of immediate multiactions). To resolve conflicts among deterministic multiactions, they are additionally equipped with positive real-valued weights. As argued in [43–45], a combination of deterministic and stochastic delays fits well to model technical systems with constant (fixed) durations of the regular non-random activities and probabilistically distributed (stochastic) durations of the randomly occurring activities. dtsdPBC has a step operational semantics, defined via labeled probabilistic transition systems. The denotational semantics of dtsdPBC was defined in terms of a subclass of labeled discrete time stochastic and deterministic Petri nets (LDTSDPNs), called dtsd-boxes.

1.4. Our framework

As a basis model, we take *discrete time stochastic and deterministic Petri box calculus* (dtsdPBC) [38–42], featuring a step operational semantics. Here we do not consider

the Petri net denotational semantics of the calculus, since it was extensively described in [39]. In that paper, a consistency of the operational and denotational semantics with respect to step stochastic bisimulation equivalence was proved. Hence, all the results established for the former can be readily transferred to the latter up to that equivalence.

In [40,42], with the *embedding* method, based on the embedded DTMC (EDTMC) specifying the state change probabilities, we constructed and solved the underlying stochastic process, which is a semi-Markov chain (SMC). The obtained stationary probability masses and average sojourn times in the states of the SMC were used to calculate the performance measures (indices) of interest. The alternative solution techniques were also developed, called *abstraction* and *elimination*, that are based respectively on the corresponding discrete time Markov chain (DTMC) and its reduction (RDTMC) by eliminating vanishing states (those with zero sojourn times).

In [39,41], we proposed step stochastic bisimulation equivalence to identify the algebraic processes with similar qualitative and quantitative behavior. We established consistency of the operational and denotational semantics of dtsdPBC up to that equivalence. We examined its interrelations with other equivalences of the algebra. The introduced equivalence was applied to reduce (by quotienting) the transition systems, SMCs, DTMCs and RDTMCs of the process expressions while preserving their qualitative and quantitative characteristics. We proved that the equivalence guarantees identity of the stationary behavior and residence time properties in the equivalence classes. This implies coincidence of the performance indices based on the steady-state probabilities and sojourn time averages for the complete and quotient behaviour.

The main advantages of dtsdPBC are flexible multiaction labels, stochastic and deterministic multiactions, powerful operations, as well as a step operational and a Petri net denotational semantics allowing for concurrent execution of activities (transitions), with an ability for analytical and parametric performance evaluation. The uniqueness of this approach consists in applying a parallel semantics for the expressions and preserving the concurrency level in the extracted performance models (SMC, DTMC and RDTMC) through their state changes corresponding to the simultaneous executions.

1.5. Our contributions

In the present paper, we describe a case study of a system consisting of two processors and a common shared memory with maintenance that explains how to model concurrent systems within the calculus and analyze their performance, as well as how to reduce the systems behavior while preserving their performance indices and making easier the performance evaluation. We study a generalized variant of the shared memory system by treating the probabilities and weights from the standard system's specification as variables (parameters) that possess general values.

First, we consider a process expression of the concrete system that differentiates among the processors, and then we analyze its functionality and performance using the corresponding transition system, SMC, DTMC and RDTMC. Second, we model the abstract system that abstracts from the names of the processors (by making identical the actions from their specifications), in order to apply reduction by the equivalence. The quotients of the abstract system's behavior (represented by the transition system, SMC, DTMC and RDTMC) by the step stochastic bisimulation equivalence are constructed. Third, the generalized probabilities of the reduced quotient DTMC (coinciding with the quotient RDTMC) are treated as parameters (variables of the performance index functions) to be adjusted for the abstract system's performance optimization.

Thus, the main contributions of the paper are as follows.

- Performance analysis of the concrete generalized shared memory system with maintenance using its transition system, SMC, DTMC and RDTMC.
- Performance analysis of the abstract system via its quotient (by step stochastic bisimulation equivalence) transition system, SMC, DTMC and RDTMC.
- Performance optimization by adjusting the quotient RDTMC probabilities, treated as parameters of the abstract system's performance index functions.

1.6. Structure of the paper

In Section 2, the syntax of algebra dtsdPBC is proposed. In Section 3, the operational semantics of the calculus in terms of labeled probabilistic transition systems is presented. Step stochastic bisimulation equivalence is defined and investigated in Section 4. In Section 5, the equivalence quotients of the transition systems and corresponding Markov chains of the process expressions are constructed. In Section 6, the introduced equivalence is proved to preserve the stationary behavior and residence time properties in the equivalence classes. In Section 7, the generalized shared memory system with maintenance is presented as a case study. Section 8 discusses the results obtained and outlines research perspectives in this area.

2. Syntax

In this section, we propose the syntax: activities, operations and expressions.

2.1. Activities and operations

Multiset is a set with allowed identical elements.

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

We denote the *set of all finite multisets* over a set X by $\mathbb{N}_{fin}^X$. Let $M, M' \in \mathbb{N}_{fin}^X$. The *cardinality* of M is $|M| = \sum_{x \in X} M(x)$. We write $x \in M$ if $M(x) > 0$ and $M \subseteq M'$ if $\forall x \in X \ M(x) \leq M'(x)$. We define $(M + M')(x) = M(x) + M'(x)$ and $(M - M')(x) = \max\{0, M(x) - M'(x)\}$. When $\forall x \in X, \ M(x) \leq 1, \ M$ can be seen as a proper set $M \subseteq X$. The *set of all subsets (powerset)* of X is denoted by 2^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 $\mathcal{L} = \mathbb{N}_{fin}^{\mathcal{A}}$ be the set of *all multiactions*. Then $\emptyset \in \mathcal{L}$ specifies an internal move, i.e. the execution of a multiaction without visible action names. The *alphabet* of $\alpha \in \mathcal{L}$ is defined as $\mathcal{A}(\alpha) = \{x \in \mathcal{A} \mid \alpha(x) > 0\}$.

A *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) sets of stochastic multiactions after one time unit delay. The probability 1 is left for (implicitly assigned to) waiting multiactions, i.e. positively delayed deterministic multiactions (to be defined later), which have weights to resolve conflicts with other waiting multiactions. Let $\mathcal{SL}$ be the

set of *all stochastic multiactions*.

A *deterministic multiaction* is a pair $(\alpha, \natural_l^\theta)$, where $\alpha \in \mathcal{L}$, $\theta \in \mathbb{N}$ is the non-negative integer-valued *(fixed) delay* and $l \in \mathbb{R}_{>0} = (0; \infty)$ is the positive real-valued *weight* of the multiaction α. This weight is interpreted as a measure of importance (urgency, interest) or a bonus reward associated with execution of the deterministic multiaction at the moment when the corresponding delay has expired. Such weights are used to calculate the probabilities to execute sets of deterministic multiactions after their delays. An *immediate multiaction* is a deterministic multiaction with the delay 0 while a *waiting multiaction* is a deterministic multiaction with a positive delay. In case of no conflicts among waiting multiactions, whose remaining times to execute (RTEs) are equal to one time unit, they are executed with probability 1 at the next moment. Deterministic multiactions have a priority over stochastic ones while immediate multiactions have a priority over waiting ones. Different types of multiactions cannot participate together in some step (parallel execution). Let $\mathcal{DL}$ be the set of *all deterministic multiactions*, $\mathcal{IL}$ be the set of *all immediate multiactions* and $\mathcal{WL}$ be the set of *all waiting multiactions*. We have $\mathcal{DL} = \mathcal{IL} \cup \mathcal{WL}$.

The same multiaction $\alpha \in \mathcal{L}$ may have different probabilities, (fixed) delays and weights in the same specification. An *activity* is a stochastic or a deterministic multiaction. Let $\mathcal{SDL} = \mathcal{SL} \cup \mathcal{DL} = \mathcal{SL} \cup \mathcal{IL} \cup \mathcal{WL}$ be the set of *all activities*. The *alphabet* of an activity $(\alpha, \kappa) \in \mathcal{SDL}$ is defined as $\mathcal{A}(\alpha, \kappa) = \mathcal{A}(\alpha)$. The *alphabet* of a multiset of activities $\Upsilon \in \mathbb{N}_{fin}^{\mathcal{SDL}}$ is defined as $\mathcal{A}(\Upsilon) = \cup_{(\alpha,\kappa)\in\Upsilon}\mathcal{A}(\alpha)$.

Activities are combined into formulas (process expressions) by the following operations: *sequence* ;, *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.

Sequence (sequential composition) and choice (composition) have a standard interpretation, like in other process algebras, but parallelism (parallel composition) does not include synchronization, unlike that operation in CCS [3].

Relabeling functions $f : \mathcal{A} \to \mathcal{A}$ are bijections preserving conjugates, i.e. $\forall x \in \mathcal{A}\ f(\hat{x}) = \widehat{f(x)}$. Relabeling is extended to multiactions: for $\alpha \in \mathcal{L}$ we define $f(\alpha) = \sum_{x\in\alpha} f(x)$. Relabeling is extended to activities: for $(\alpha, \kappa) \in \mathcal{SDL}$, we define $f(\alpha, \kappa) = (f(\alpha), \kappa)$. Relabeling is extended to the multisets of activities: for $\Upsilon \in \mathbb{N}_{fin}^{\mathcal{SDL}}$ we define $f(\Upsilon) = \sum_{(\alpha,\kappa)\in\Upsilon}(f(\alpha), \kappa)$.

Restriction over an elementary action $a \in Act$ means that, for a given expression, any process behavior 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 $(\alpha \oplus_a \beta)(x) = \begin{cases} \alpha(x) + \beta(x) - 1, & x = a \text{ or } x = \hat{a}; \\ \alpha(x) + \beta(x), & \text{otherwise.} \end{cases}$

Activities are synchronized via their multiaction parts, i.e. the synchronization by a of two activities, whose multiaction parts α and β possess the properties mentioned above, results in the activity with the multiaction part $\alpha \oplus_a \beta$. We may synchronize activities of the same type only: either both stochastic multiactions or both deterministic ones *with the same delay*, since stochastic, waiting and immediate multiactions have different priorities, and diverse delays of waiting multiactions would contradict their joint timing. Note that the execution of immediate multiactions takes no time, unlike that of waiting or stochastic ones. Synchronization by a means that, for a given expression with a process behavior containing two concurrent activities that can be synchronized by a, there exists also the behavior that differs from the former only in

that the two activities are replaced by the result of their synchronization.

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

2.2. Process expressions

Static expressions specify the structure of processes, i.e. how activities are combined by operations to construct the composite process-algebraic formulas. As for the PN intuition, static expressions correspond to unmarked LDTSDPNs [38,39]. A marking is the allocation of tokens in the places of a PN. Markings are used to describe dynamic behavior of PNs in terms of transition firings.

We assume that every waiting multiaction has a countdown timer associated, whose value is the time left till the moment when the waiting multiaction can be executed. Thus, besides standard (unstamped) waiting multiactions $(\alpha, \natural_l^\theta) \in \mathcal{WL}$, a special case of the *stamped* waiting multiactions should be considered in the definition of static expressions. Each (time) stamped waiting multiaction $(\alpha, \natural_l^\theta)^\delta$ has an extra superscript $\delta \in \{1, \ldots, \theta\}$ that specifies a time stamp indicating the *latest* value of the timer associated with that multiaction. The standard waiting multiactions have no time stamps, to show irrelevance of the timer values for them (for example, their timers have not yet started or have already finished). The alphabet part for (the multisets of) stamped waiting multiactions is defined like that for (the multisets of) unstamped ones.

For simplicity, we do not assign the timer value superscripts δ to immediate multiactions, a special case of deterministic multiactions $(\alpha, \natural_l^\theta)$ with the delay $\theta = 0$ in the form of $(\alpha, \natural_l^0)$, since their timer values can only be equal to 0.

Definition 2.2. Let $(\alpha, \kappa) \in \mathcal{SDL}$, $(\alpha, \natural_l^\theta) \in \mathcal{WL}$, $\delta \in \{1, \ldots, \theta\}$ and $a \in Act$. A *static expression* of dtsdPBC is

$$E ::= (\alpha, \kappa) \mid (\alpha, \natural_l^\theta)^\delta \mid E; E \mid E[]E \mid E \| E \mid E[f] \mid E \ \mathsf{rs}\ a \mid E \ \mathsf{sy}\ a \mid [E * E * E].$$

Let *StatExpr* denote the set of *all static expressions* of dtsdPBC.

To avoid technical difficulties with the iteration operator, we should not allow concurrency at the highest level of the second argument of iteration. This is not a severe restriction, since we can always prefix parallel expressions by an activity with the empty multiaction part. Relaxing the restriction can result in LDTSDPNs [38,39] which are not safe, like shown for PNs in [9]. A PN is *n-bounded* ($n \in \mathbb{N}$) 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 $(\alpha, \kappa) \in \mathcal{SDL}$, $(\alpha, \natural_l^\theta) \in \mathcal{WL}$, $\delta \in \{1, \ldots, \theta\}$ and $a \in Act$. A *regular static expression* of dtsdPBC is

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

Let *RegStatExpr* denote the set of *all regular static expressions* of dtsdPBC.

Let E be a regular static expression. The *underlying timer-free regular static expression* $\lfloor E$ of E is obtained by removing from it all timer value superscripts.

The set of *all stochastic multiactions (from the syntax) of* E is $\mathcal{SL}(E) = \{(\alpha, \rho) \mid (\alpha, \rho)$ is a subexpression of $E\}$. The set of *all immediate multiactions (from the syn-*

tax) of E is $\mathcal{IL}(E) = \{(\alpha, \natural_l^0) \mid (\alpha, \natural_l^0)$ is a subexpression of $E\}$. The set of *all waiting multiactions (from the syntax) of* E is $\mathcal{WL}(E) = \{(\alpha, \natural_l^\theta) \mid (\alpha, \natural_l^\theta)$ or $(\alpha, \natural_l^\theta)^\delta$ is a subexpression of E for $\delta \in \{1, \ldots, \theta\}\}$. Thus, the set of *all deterministic multiactions (from the syntax) of* E is $\mathcal{DL}(E) = \mathcal{IL}(E) \cup \mathcal{WL}(E)$ and the set of *all activities (from the syntax) of* E is $\mathcal{SDL}(E) = \mathcal{SL}(E) \cup \mathcal{DL}(E) = \mathcal{SL}(E) \cup \mathcal{IL}(E) \cup \mathcal{WL}(E)$.

Dynamic expressions specify the states of processes, i.e. particular stages of the process behavior. As for the Petri net intuition, dynamic expressions correspond to marked LDTSDPNs [38,39]. Dynamic expressions are obtained from static ones, by annotating them with upper or lower bars which specify the active components of the system at the current moment of time. 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.

For every overlined stamped waiting multiaction $\overline{(\alpha, \natural_l^\theta)^\delta}$, the superscript $\delta \in \{1, \ldots, \theta\}$ specifies the *current* value of the *running* countdown timer associated with the waiting multiaction. That decreasing discrete timer is started with the *initial* value θ (the waiting multiaction delay) at the moment when the waiting multiaction becomes overlined. Then such a newly overlined stamped waiting multiaction $\overline{(\alpha, \natural_l^\theta)^\theta}$ is similar to the freshly overlined unstamped waiting multiaction $\overline{(\alpha, \natural_l^\theta)}$. Such similarity will be captured by the structural equivalence, defined later.

While the stamped waiting multiaction stays overlined with the process execution, the timer decrements by one discrete time unit with each global time tick until the timer value becomes 1. This means that one unit of time remains till execution of that multiaction (the remaining time to execute, RTE, equals one). Its execution should follow in the next moment with probability 1, in case there are no conflicting with it immediate multiactions or conflicting waiting multiactions whose RTEs equal to one, and it is not affected by restriction. An activity is affected by restriction, if it is within the scope of a restriction operation with the argument action, such that it or its conjugate is contained in the multiaction part of that activity.

Definition 2.4. Let $E \in StatExpr$ and $a \in Act$. A *dynamic expression* of dtsdPBC is

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

Let $DynExpr$ denote the set of *all dynamic expressions* of dtsdPBC.

Let G be a dynamic expression. The *underlying static (line-free) expression* $\lfloor G \rfloor$ of G is obtained by removing from it all upper and lower bars. If the underlying static expression of a dynamic one is not regular, the corresponding LDTSDPN can be non-safe [38,39] (2-bounded in the worst case, like for PNs [9]).

Definition 2.5. A dynamic expression G is *regular* if $\lfloor G \rfloor$ is regular.

Let $RegDynExpr$ denote the set of *all regular dynamic expressions* of dtsdPBC.

Let G be a regular dynamic expression. The *underlying timer-free regular dynamic expression* $\lfloor G$ is obtained by removing from G all timer value superscripts.

The set of *all stochastic (immediate or waiting, respectively) multiactions (from the syntax) of* G is defined as $\mathcal{SL}(G) = \mathcal{SL}(\lfloor G \rfloor)$ ($\mathcal{IL}(G) = \mathcal{IL}(\lfloor G \rfloor)$ or $\mathcal{WL}(G) = \mathcal{WL}(\lfloor G \rfloor)$, respectively). Thus, the set of *all deterministic multiactions (from the syntax) of* G is $\mathcal{DL}(G) = \mathcal{IL}(G) \cup \mathcal{WL}(G)$ and the set of *all activities (from the syntax) of* G is $\mathcal{SDL}(G) = \mathcal{SL}(G) \cup \mathcal{DL}(G) = \mathcal{SL}(G) \cup \mathcal{IL}(G) \cup \mathcal{WL}(G)$.

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

$\overline{(\alpha, \natural_l^\theta)} \Rightarrow \overline{(\alpha, \natural_l^\theta)^\theta}$	$\overline{E; F} \Rightarrow \overline{E}; F$	$\underline{E}; F \Rightarrow E; \overline{F}$
$E; \underline{F} \Rightarrow \underline{E; F}$	$\overline{E[]F} \Rightarrow \overline{E}[]F$	$\overline{E[]F} \Rightarrow E[]\overline{F}$
$\underline{E}[]F \Rightarrow \underline{E[]F}$	$E[]\underline{F} \Rightarrow \underline{E[]F}$	$\overline{E \| F} \Rightarrow \overline{E} \| \overline{F}$
$\underline{E} \| \underline{F} \Rightarrow \underline{E \| F}$	$\overline{E[f]} \Rightarrow \overline{E}[f]$	$\underline{E}[f] \Rightarrow \underline{E[f]}$
$\overline{E \ \mathsf{rs}\ a} \Rightarrow \overline{E} \ \mathsf{rs}\ a$	$\underline{E} \ \mathsf{rs}\ a \Rightarrow \underline{E \ \mathsf{rs}\ a}$	$\overline{E \ \mathsf{sy}\ a} \Rightarrow \overline{E} \ \mathsf{sy}\ a$
$\underline{E} \ \mathsf{sy}\ a \Rightarrow \underline{E \ \mathsf{sy}\ a}$	$\overline{[E * F * K]} \Rightarrow [\overline{E} * F * K]$	$[\underline{E} * F * K] \Rightarrow [E * \overline{F} * K]$
$[E * \underline{F} * K] \Rightarrow [E * \overline{F} * K]$	$[E * \underline{F} * K] \Rightarrow [E * F * \overline{K}]$	$[E * F * \underline{K}] \Rightarrow \underline{[E * F * K]}$

Table 2. Inaction rules for arbitrary regular dynamic expressions.

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

3. Operational semantics

In this section, we define the operational semantics via labeled transition systems.

3.1. Inaction rules

The inaction rules for dynamic expressions describe their structural transformations in the form of $G \Rightarrow \widetilde{G}$ which do not change the states of the specified processes. The goal of those syntactic transformations is to obtain the well-structured resulting expressions called operative ones to which no inaction rules can be further applied. The application of an inaction rule to a dynamic expression does not lead to time tick or transition firing in the corresponding LDTSDPN [38,39], hence, its current marking stays unchanged.

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

In Table 1, we define inaction rules for regular dynamic expressions being overlined and underlined static ones. In this table, $(\alpha, \natural_l^\theta) \in \mathcal{WL},\ \delta \in \{1, \ldots, \theta\},\ E, F, K \in RegStatExpr$ and $a \in Act$. The first inaction rule suggests that the timer value of each newly overlined waiting multiaction is set to the delay of it.

In Table 2, we introduce inaction rules for regular dynamic expressions in the arbitrary form. In this table, $E, F \in RegStatExpr,\ G, H, \widetilde{G}, \widetilde{H} \in RegDynExpr$ and $a \in Act$. For brevity, two distinct inaction rules with the same premises are collated in some cases, resulting in the inaction rules with double conclusion.

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

Let $OpRegDynExpr$ denote the set of *all operative regular dynamic expressions* of dtsdPBC. Note that any dynamic expression can be always transformed into a (not

necessarily unique) operative one by using the inaction rules. In the following, we consider regular expressions only and omit the word 'regular'.

Definition 3.2. The relation $\approx = (\Rightarrow \cup \Leftarrow)^*$ is a *structural equivalence* of dynamic expressions in dtsdPBC. 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 a backward direction.

Let G be a dynamic expression. Then $[G]_\approx = \{H \mid G \approx H\}$ is the equivalence class of G with respect to the structural equivalence, called the (corresponding) *state*. Next, G is an *initial* dynamic expression, denoted by $init(G)$, if $\exists E \in RegStatExpr\ G \in [\overline{E}]_\approx$. Further, G is a *final* dynamic expression, denoted by $final(G)$, if $\exists E \in RegStatExpr\ G \in [\underline{E}]_\approx$.

Let G be a dynamic expression and $s = [G]_\approx$. The set of *all enabled stochastic multiactions of* s is $EnaSto(s) = \{(\alpha, \rho) \in \mathcal{SL} \mid \exists H \in s \cap OpRegDynExpr\ \overline{(\alpha, \rho)}$ is a subexpression of $H\}$. The set of *all enabled immediate multiactions of* s is $EnaImm(s) = \{(\alpha, \natural_l^0) \in \mathcal{IL} \mid \exists H \in s \cap OpRegDynExpr\ \overline{(\alpha, \natural_l^0)}$ is a subexpression of $H\}$. The set of *all enabled waiting multiactions of* s is $EnaWait(s) = \{(\alpha, \natural_l^\theta) \in \mathcal{WL} \mid \exists H \in s \cap OpRegDynExpr\ \overline{(\alpha, \natural_l^\theta)^\delta},\ \delta \in \{1, \ldots, \theta\},$ is a subexpression of $H\}$. The set of *all newly enabled waiting multiactions of* s is $EnaWaitNew(s) = \{(\alpha, \natural_l^\theta) \in \mathcal{WL} \mid \exists H \in s \cap OpRegDynExpr\ \overline{(\alpha, \natural_l^\theta)^\theta}$ is a subexpression of $H\}$.

The set of *all enabled deterministic multiactions of* s is $EnaDet(s) = EnaImm(s) \cup EnaWait(s)$ and the set of *all enabled activities of* s is $Ena(s) = EnaSto(s) \cup EnaDet(s) = EnaSto(s) \cup EnaImm(s) \cup EnaWait(s)$. Then $Ena(s) = Ena([G]_\approx)$ is an algebraic analogue of the set of all transitions enabled at the initial marking of the LDTSDPN [38,39] corresponding to G. The activities, resulted from synchronization, are not present in the syntax of the dynamic expressions. Their enabledness status can be recovered by observing that of the pair of synchronized activities from the syntax (they both should be enabled for enabling their synchronous product), even if they are affected by restriction after the synchronization.

Definition 3.3. An operative dynamic expression G is *saturated (with the values of timers)*, if each enabled waiting multiaction of $[G]_\approx$, being superscribed with the value of its timer and possibly overlined, is the subexpression of G.

Let $SaOpRegDynExpr$ denote the set of *all saturated operative dynamic expressions* of dtsdPBC.

Proposition 3.4 ([38,39])**.** *Any operative dynamic expression can be transformed into the saturated one by applying the inaction rules in a forward or a backward direction.*

Thus, any dynamic expression can be transformed into a (not necessarily unique) saturated operative one by (possibly reverse) applying the inaction rules.

Let G be a saturated operative dynamic expression. Then $\circlearrowleft G$ denotes the *timer decrement* operator $\circlearrowleft$, applied to G. The result is a saturated operative dynamic expression, obtained from G via decrementing by one all greater than 1 values of the timers associated with all (if any) stamped waiting multiactions from the syntax of G. Each such stamped waiting multiaction changes its timer value from $\delta \in \mathbb{N}_{\geq 1}$ in G to $\max\{1, \delta - 1\}$ in $\circlearrowleft G$. The timer decrement operator affects the (possibly overlined or underlined) stamped waiting multiactions being the subexpressions of G as: $(\alpha, \natural_l^\theta)^\delta$ is replaced with $(\alpha, \natural_l^\theta)^{\max\{1, \delta-1\}}$, and similarly for the overlined or underlined ones.

Note that when $\delta = 1$, we have $\max\{1, \delta - 1\} = \max\{1, 0\} = 1$, hence, the timer value $\delta = 1$ may remain unchanged for a stamped waiting multiaction that is not executed by some reason at the next time moment, but stays stamped. For example, that stamped waiting multiaction may be affected by restriction. If the timer values cannot be decremented with a time tick for all stamped waiting multiactions (if any) from G then $\circlearrowleft G = G$ and we obtain so-called *empty loop* transition, defined later.

The timer decrement operator keeps stamping of the waiting multiactions, since it may only decrease their timer values, so that the stamped waiting multiactions stay stamped (with their timer values, possibly decremented by one).

3.2. Action and empty move rules

The action rules are applied when some activities are executed. With these rules we capture the prioritization among different types of multiactions. We also have the empty move rule, used to capture a delay of one discrete time unit when no immediate or waiting multiactions are executable. In this case, the empty multiset of activities is executed. The action and empty move 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 or delays and weights of the activities to be executed from the current process state will be used to calculate the probabilities of such executions.

The action rules with stochastic (immediate or waiting, respectively) multiactions describe dynamic expression transformations in the form of $G \stackrel{\Gamma}{\to} \widetilde{G}$ ($G \stackrel{I}{\to} \widetilde{G}$ or $G \stackrel{W}{\to} \widetilde{G}$, respectively) due to execution of non-empty multisets Γ of stochastic (I of immediate or W of waiting, respectively) multiactions. The rules represent possible state changes of the specified processes when some non-empty multisets of stochastic (immediate or waiting, respectively) multiactions are executed. The application of an action rule with stochastic (immediate or waiting, respectively) multiactions to a dynamic expression leads in the corresponding LDTSDPN [38,39] to a discrete time tick at which some stochastic or waiting transitions fire (or to the instantaneous firing of some immediate transitions) and possible change of the current marking. The current marking stays unchanged only if there is a self-loop produced by the iterative execution of a non-empty multiset, which must be one-element, since we allow no concurrency at the highest level of the second argument of iteration.

The empty move rule (applicable only when no immediate or waiting multiactions can be executed from the current state) describes dynamic expression transformations in the form of $G \stackrel{\emptyset}{\to} \circlearrowleft G$, called the *empty moves*, due to execution of the empty multiset of activities at a discrete time tick. When no timer values are decremented within G with the empty multiset execution at the next moment (for example, if G contains no stamped waiting multiactions), we have $\circlearrowleft G = G$. In such a case, the empty move from G is in the form of $G \stackrel{\emptyset}{\to} G$, called the *empty loop*. The application of the empty move rule to a dynamic expression leads to a discrete time tick in the corresponding LDTSDPN [38,39] at which no transitions fire and the current marking is not changed, but the timer values of the waiting transitions enabled at the marking (if any) are decremented by one. This is a new rule that has no prototype among inaction rules of PBC, since it represents a time delay.

Thus, an application of every action rule with stochastic or waiting multiactions or the empty move rule requires one discrete time unit delay, i.e. the execution of a (possibly empty) multiset of stochastic or (non-empty) multiset of waiting multi-

actions leading to the dynamic expression transformation described by the rule is accomplished instantly after one time unit. An application of every action rule with immediate multiactions does not take any time, i.e. the execution of a (non-empty) multiset of immediate multiactions is accomplished instantly at the current moment.

The expressions of dtsdPBC can contain identical activities. To avoid technical difficulties, such as calculation of the probabilities for multiple transitions, we can enumerate coinciding activities 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 now define the numbering which encodes a binary tree with the leaves labeled by natural numbers.

Definition 3.5. The *numbering* of expressions is $\iota ::= \ n \mid (\iota)(\iota)$, where $n \in \mathbb{N}$.

Let Num denote the set of *all numberings* of expressions.

The new activities resulting from synchronizations 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 identify the mentioned instances. The *content* of a numbering $\iota \in Num$ is $Cont(\iota) = \begin{cases} \{\iota\}, & \iota \in \mathbb{N}; \\ Cont(\iota_1) \cup Cont(\iota_2), & \iota = (\iota_1)(\iota_2). \end{cases}$
After the enumeration, the multisets of activities from the expressions become the proper sets. We suppose that the identical activities are enumerated when needed to avoid ambiguity. This enumeration is considered to be implicit.

Definition 3.6. Let $G \in OpRegDynExpr$. We define the *set of all non-empty multisets of activities which can be potentially executed from* G, denoted by $Can(G)$. Let $(\alpha, \kappa) \in \mathcal{SDL},\ E, F \in RegStatExpr,\ H \in OpRegDynExpr$ and $a \in Act$.

(1) If $final(G)$ then $Can(G) = \emptyset$.
(2) If $G = \overline{(\alpha, \kappa)^\delta}$ and $\kappa = \natural_l^\theta,\ \theta \in \mathbb{N}_{\geq 2},\ l \in \mathbb{R}_{>0},\ \delta \in \{2, \ldots, \theta\}$, then $Can(G) = \emptyset$.
(3) If $G = \overline{(\alpha, \kappa)}$ and $\kappa \in (0; 1)$ or $\kappa = \natural_l^0,\ l \in \mathbb{R}_{>0}$, then $Can(G) = \{\{(\alpha, \kappa)\}\}$.
(4) If $G = \overline{(\alpha, \kappa)^1}$ and $\kappa = \natural_l^\theta,\ \theta \in \mathbb{N}_{\geq 1},\ l \in \mathbb{R}_{>0}$, then $Can(G) = \{\{(\alpha, \kappa)\}\}$.
(5) If $\Upsilon \in Can(G)$ then $\Upsilon \in Can(G \circ E),\ \Upsilon \in Can(E \circ G)\ (\circ \in \{;, []\})$,
$\Upsilon \in Can(G \| H),\ \Upsilon \in Can(H \| G),\ f(\Upsilon) \in Can(G[f]),\ \Upsilon \in Can(G\ \mathsf{rs}\ a)$
(when $a, \hat{a} \notin \mathcal{A}(\Upsilon)$), $\Upsilon \in Can(G\ \mathsf{sy}\ a),\ \Upsilon \in Can([G * E * F])$,
$\Upsilon \in Can([E * G * F]),\ \Upsilon \in Can([E * F * G])$.
(6) If $\Upsilon \in Can(G)$ and $\Xi \in Can(H)$ then $\Upsilon + \Xi \in Can(G \| H)$.
(7) If $\Upsilon \in Can(G\ \mathsf{sy}\ a)$ and $(\alpha, \kappa), (\beta, \lambda) \in \Upsilon$ are different, $a \in \alpha,\ \hat{a} \in \beta$, then
   (a) $\Upsilon - \{(\alpha, \kappa), (\beta, \lambda)\} + \{(\alpha \oplus_a \beta, \kappa \cdot \lambda)\} \in Can(G\ \mathsf{sy}\ a)$ if $\kappa, \lambda \in (0; 1)$;
   (b) $\Upsilon - \{(\alpha, \kappa), (\beta, \lambda)\} + \{(\alpha \oplus_a \beta, \natural_{l+m}^\theta)\} \in Can(G\ \mathsf{sy}\ a)$ if $\kappa = \natural_l^\theta,\ \lambda = \natural_m^\theta$, $\theta \in \mathbb{N},\ l, m \in \mathbb{R}_{>0}$.

When we synchronize the same multiset of activities in different orders, we obtain several activities with the same multiaction and probability or delay and weight parts, but with different numberings having the same content. Then we only consider a single one of the resulting activities.

If $\Upsilon \in Can(G)$ then by definition of $Can(G),\ \forall \Xi \subseteq \Upsilon,\ \Xi \neq \emptyset$, we have $\Xi \in Can(G)$.

Let $G \in OpRegDynExpr$ and $Can(G) \neq \emptyset$. Obviously, if there are only stochastic (immediate or waiting, respectively) multiactions in the multisets from $Can(G)$ then these stochastic (immediate or waiting, respectively) multiactions can be exe-

cuted from G. Otherwise, besides stochastic ones, there are also deterministic (immediate and/or waiting) multiactions in the multisets from $Can(G)$. By the note above, there are non-empty multisets of deterministic multiactions in $Can(G)$ as well, i.e. $\exists \Upsilon \in Can(G)\ \Upsilon \in \mathbb{N}_{fin}^{\mathcal{DL}} \setminus \{\emptyset\}$. In this case, no stochastic multiactions can be executed from G, even if $Can(G)$ contains non-empty multisets of stochastic multiactions, since deterministic multiactions have a priority over stochastic ones, and should be executed first. Further, if there are no stochastic, but both waiting and immediate multiactions in the multisets from $Can(G)$, then, analogously, no waiting multiactions can be executed from G, since immediate multiactions have a priority over waiting ones.

When there are only waiting and, possibly, stochastic multiactions in the multisets from $Can(G)$ then only waiting ones can be executed from G. Then just *maximal* non-empty multisets of waiting multiactions can be executed from G, since all non-conflicting waiting multiactions cannot wait and they should occur at the next time moment with probability 1. The next definition formalizes these requirements.

Definition 3.7. Let $G \in OpRegDynExpr$. The *set of all non-empty multisets of activities which can be executed from* G is

$$Now(G) = \begin{cases} Can(G) \cap \mathbb{N}_{fin}^{\mathcal{IL}}, & Can(G) \cap \mathbb{N}_{fin}^{\mathcal{IL}} \neq \emptyset; \\ \{W \in Can(G) \cap \mathbb{N}_{fin}^{\mathcal{WL}} \mid & (Can(G) \cap \mathbb{N}_{fin}^{\mathcal{IL}} = \emptyset) \wedge \\ \forall V \in Can(G) \cap \mathbb{N}_{fin}^{\mathcal{WL}}\ W \subseteq V\ \Rightarrow\ V = W\}, & (Can(G) \cap \mathbb{N}_{fin}^{\mathcal{WL}} \neq \emptyset); \\ Can(G), & \text{otherwise.} \end{cases}$$

Let $G \in OpRegDynExpr$. The expression G is *s-tangible (stochastically tangible)*, denoted by $stang(G)$, if $Now(G) \subseteq \mathbb{N}_{fin}^{\mathcal{SL}} \setminus \{\emptyset\}$. In particular, we have $stang(G)$, if $Now(G) = \emptyset$. The expression G is *w-tangible (waitingly tangible)*, denoted by $wtang(G)$, if $\emptyset \neq Now(G) \subseteq \mathbb{N}_{fin}^{\mathcal{WL}} \setminus \{\emptyset\}$. The expression G is *tangible*, denoted by $tang(G)$, if $stang(G)$ or $wtang(G)$, i.e. $Now(G) \subseteq (\mathbb{N}_{fin}^{\mathcal{SL}} \cup \mathbb{N}_{fin}^{\mathcal{WL}}) \setminus \{\emptyset\}$. Again, we particularly have $tang(G)$, if $Now(G) = \emptyset$. Otherwise, the expression G is *vanishing*, denoted by $vanish(G)$, and in this case $\emptyset \neq Now(G) \subseteq \mathbb{N}_{fin}^{\mathcal{IL}} \setminus \{\emptyset\}$. Note that the operative dynamic expressions from $[G]_{\approx}$ may have different types in general.

Let $G \in RegDynExpr$. We write $stang([G]_{\approx})$, if $\forall H \in [G]_{\approx} \cap OpRegDynExpr\ stang(H)$. We write $wtang([G]_{\approx})$, if $\exists H \in [G]_{\approx} \cap OpRegDynExpr\ wtang(H)$ and $\forall H' \in [G]_{\approx} \cap OpRegDynExpr\ tang(H')$. We write $tang([G]_{\approx})$, if $stang([G]_{\approx})$ or $wtang([G]_{\approx})$. Otherwise, we write $vanish([G]_{\approx})$, and in this case $\exists H \in [G]_{\approx} \cap OpRegDynExpr\ vanish(H)$.

In Table 3, we define the action and empty move rules. In the table, $(\alpha, \rho), (\beta, \chi) \in \mathcal{SL}$, $(\alpha, \natural_l^0), (\beta, \natural_m^0) \in \mathcal{IL}$ and $(\alpha, \natural_l^\theta), (\beta, \natural_m^\theta) \in \mathcal{WL}$. Further, $E, F \in RegStatExpr$, $G, H \in SatOpRegDynExpr$, $\widetilde{G}, \widetilde{H} \in RegDynExpr$ and $a \in Act$. Next, $\Gamma, \Delta \in \mathbb{N}_{fin}^{\mathcal{SL}} \setminus \{\emptyset\}$, $\Gamma' \in \mathbb{N}_{fin}^{\mathcal{SL}}$, $I, J \in \mathbb{N}_{fin}^{\mathcal{IL}} \setminus \{\emptyset\}$, $I' \in \mathbb{N}_{fin}^{\mathcal{IL}}$, $V, W \in \mathbb{N}_{fin}^{\mathcal{WL}} \setminus \{\emptyset\}$, $V' \in \mathbb{N}_{fin}^{\mathcal{WL}}$ and $\Upsilon \in \mathbb{N}_{fin}^{\mathcal{SDL}} \setminus \{\emptyset\}$.

We use the next abbreviations in the names of the rules: '**E**' for '**E**mpty move', '**B**' for '**B**asis case', '**S**' for '**S**equence', '**C**' for '**C**hoice', '**P**' for '**P**arallel', '**L**' for 're**L**abeling', '**R**' for '**R**estriction', '**I**' for '**I**teraton' and '**Sy**' for '**Sy**nchronization'. The first rule is the empty move rule **E**. The other rules are the action rules, describing transformations of dynamic expressions, which are built using particular algebraic operations. If we cannot merge the rules with stochastic, immediate ans waiting multiactions in one rule for some operation then we get the coupled action rules. In such cases, the names of the action rules with stochastic multiactions have a suffix '**s**', those

Table 3. Action and empty move rules.

$$\mathbf{E}\ \frac{stang([G]_{\approx})}{G \xrightarrow{\emptyset} \circlearrowleft G} \qquad \mathbf{Bs}\ \overline{(\alpha,\rho)} \xrightarrow{\{(\alpha,\rho)\}} \underline{(\alpha,\rho)} \qquad \mathbf{Bi}\ \overline{(\alpha,\natural_l^0)} \xrightarrow{\{(\alpha,\natural_l^0)\}} \underline{(\alpha,\natural_l^0)} \qquad \mathbf{Bw}\ \overline{(\alpha,\natural_l^\theta)^1} \xrightarrow{\{(\alpha,\natural_l^\theta)\}} \underline{(\alpha,\natural_l^\theta)}$$

$$\mathbf{S}\ \frac{G \xrightarrow{\Upsilon} \widetilde{G}}{G;E \xrightarrow{\Upsilon} \widetilde{G};E,\ E;G \xrightarrow{\Upsilon} E;\widetilde{G}} \qquad \mathbf{Cs}\ \frac{G \xrightarrow{\Gamma} \widetilde{G},\ \neg init(G) \vee (init(G) \wedge stang([\overline{E}]_{\approx}))}{G[]E \xrightarrow{\Gamma} \widetilde{G}[]\lfloor E,\ E[]G \xrightarrow{\Gamma} \lfloor E[]\widetilde{G}}$$

$$\mathbf{Ci}\ \frac{G \xrightarrow{I} \widetilde{G}}{G[]E \xrightarrow{I} \widetilde{G}[]\lfloor E,\ E[]G \xrightarrow{I} \lfloor E[]\widetilde{G}} \qquad \mathbf{Cw}\ \frac{G \xrightarrow{V} \widetilde{G},\ \neg init(G) \vee (init(G) \wedge tang([\overline{E}]_{\approx}))}{G[]E \xrightarrow{V} \widetilde{G}[]\lfloor E,\ E[]G \xrightarrow{V} \lfloor E[]\widetilde{G}}$$

$$\mathbf{P1s}\ \frac{G \xrightarrow{\Gamma} \widetilde{G},\ stang([H]_{\approx})}{G\|H \xrightarrow{\Gamma} \widetilde{G}\| \circlearrowleft H,\ H\|G \xrightarrow{\Gamma} \circlearrowleft H\|\widetilde{G}} \qquad \mathbf{P1i}\ \frac{G \xrightarrow{I} \widetilde{G}}{G\|H \xrightarrow{I} \widetilde{G}\|H,\ H\|G \xrightarrow{I} H\|\widetilde{G}}$$

$$\mathbf{P1w}\ \frac{G \xrightarrow{V} \widetilde{G},\ stang([H]_{\approx})}{G\|H \xrightarrow{V} \widetilde{G}\| \circlearrowleft H,\ H\|G \xrightarrow{V} \circlearrowleft H\|\widetilde{G}} \qquad \mathbf{P2s}\ \frac{G \xrightarrow{\Gamma} \widetilde{G},\ H \xrightarrow{\Delta} \widetilde{H}}{G\|H \xrightarrow{\Gamma+\Delta} \widetilde{G}\|\widetilde{H}} \qquad \mathbf{P2i}\ \frac{G \xrightarrow{I} \widetilde{G},\ H \xrightarrow{J} \widetilde{H}}{G\|H \xrightarrow{I+J} \widetilde{G}\|\widetilde{H}}$$

$$\mathbf{P2w}\ \frac{G \xrightarrow{V} \widetilde{G},\ H \xrightarrow{W} \widetilde{H}}{G\|H \xrightarrow{V+W} \widetilde{G}\|\widetilde{H}} \qquad \mathbf{L}\ \frac{G \xrightarrow{\Upsilon} \widetilde{G}}{G[f] \xrightarrow{f(\Upsilon)} \widetilde{G}[f]} \qquad \mathbf{R}\ \frac{G \xrightarrow{\Upsilon} \widetilde{G},\ a,\hat{a} \notin \mathcal{A}(\Upsilon)}{G\ \mathsf{rs}\ a \xrightarrow{\Upsilon} \widetilde{G}\ \mathsf{rs}\ a}$$

$$\mathbf{I1}\ \frac{G \xrightarrow{\Upsilon} \widetilde{G}}{[G*E*F] \xrightarrow{\Upsilon} [\widetilde{G}*E*F]} \qquad \mathbf{I2s}\ \frac{G \xrightarrow{\Gamma} \widetilde{G},\ \neg init(G) \vee (init(G) \wedge stang([\overline{F}]_{\approx}))}{[E*G*F] \xrightarrow{\Gamma} [E*\widetilde{G}*\lfloor F],\ [E*F*G] \xrightarrow{\Gamma} [E*\lfloor F*\widetilde{G}]}$$

$$\mathbf{I2i}\ \frac{G \xrightarrow{I} \widetilde{G}}{[E*G*F] \xrightarrow{I} [E*\widetilde{G}*\lfloor F],\ [E*F*G] \xrightarrow{I} [E*\lfloor F*\widetilde{G}]}$$

$$\mathbf{I2w}\ \frac{G \xrightarrow{V} \widetilde{G},\ \neg init(G) \vee (init(G) \wedge tang([\overline{F}]_{\approx}))}{[E*G*F] \xrightarrow{V} [E*\widetilde{G}*\lfloor F],\ [E*F*G] \xrightarrow{V} [E*\lfloor F*\widetilde{G}]}$$

$$\mathbf{Sy1}\ \frac{G \xrightarrow{\Upsilon} \widetilde{G}}{G\ \mathsf{sy}\ a \xrightarrow{\Upsilon} \widetilde{G}\ \mathsf{sy}\ a} \qquad \mathbf{Sy2s}\ \frac{G\ \mathsf{sy}\ a \xrightarrow{\Gamma'+\{(\alpha,\rho)\}+\{(\beta,\chi)\}} \widetilde{G}\ \mathsf{sy}\ a,\ a \in \alpha,\ \hat{a} \in \beta}{G\ \mathsf{sy}\ a \xrightarrow{\Gamma'+\{(\alpha \oplus_a \beta, \rho \cdot \chi)\}} \widetilde{G}\ \mathsf{sy}\ a}$$

$$\mathbf{Sy2i}\ \frac{G\ \mathsf{sy}\ a \xrightarrow{I'+\{(\alpha,\natural_l^0)\}+\{(\beta,\natural_m^0)\}} \widetilde{G}\ \mathsf{sy}\ a,\ a \in \alpha,\ \hat{a} \in \beta}{G\ \mathsf{sy}\ a \xrightarrow{I'+\{(\alpha \oplus_a \beta, \natural_{l+m}^0)\}} \widetilde{G}\ \mathsf{sy}\ a}$$

$$\mathbf{Sy2w}\ \frac{G\ \mathsf{sy}\ a \xrightarrow{V'+\{(\alpha,\natural_l^\theta)\}+\{(\beta,\natural_m^\theta)\}} \widetilde{G}\ \mathsf{sy}\ a,\ a \in \alpha,\ \hat{a} \in \beta}{G\ \mathsf{sy}\ a \xrightarrow{V'+\{(\alpha \oplus_a \beta, \natural_{l+m}^\theta)\}} \widetilde{G}\ \mathsf{sy}\ a}$$

with immediate multiactions have a suffix '**i**', and those with waiting multiactions have a suffix '**w**'. The rules in Table 3 are explained in [38,39].

Notice that the timers of all waiting multiactions that lose their enabledness when a state change occurs become inactive (turned off) and their values become irrelevant while the timers of all those preserving their enabledness continue running with their stored values. Hence, we adapt the *enabling memory* policy [23,26,46,47] when the process states are changed and the enabledness of deterministic multiactions is possibly modified (immediate multiactions may be seen as those with the timers displaying a single value 0, so we do not need to store their values). Then the timer values of waiting multiactions are taken as the enabling memory variables.

Like in [10], we are interested in the dynamic expressions, inferred by applying the inaction rules (also in the reverse direction) and action rules from the overlined static expressions, such that no stamped (superscribed with the timer values) waiting multiaction is a subexpression of them. The reason is to ensure that time proceeds uniformly and only enabled waiting multiactions are stamped. We call such dynamic expressions reachable, by analogy with the reachable states of LDTSDPNs [38,39].

Definition 3.8. A dynamic expression G is *reachable*, if there exists a static expression E without timer value superscripts, such that $\overline{E} \approx G$ or $\overline{E} \approx G_0 \stackrel{\Upsilon_1}{\to} H_1 \approx G_1 \stackrel{\Upsilon_2}{\to} \ldots \stackrel{\Upsilon_n}{\to} H_n \approx G$ for some $\Upsilon_1, \ldots, \Upsilon_n \in \mathbb{N}_{fin}^{\mathcal{SDL}}$.

The next proposition considers enabledness of the stamped waiting multiactions.

Proposition 3.9 ([38,39])**.** *Let G be a reachable dynamic expression. Then only waiting multiactions from $EnaWait([G]_\approx)$ are stamped in G.*

3.3. Transition systems

We now construct labeled probabilistic transition systems associated with dynamic expressions, in order to subsequently define their operational semantics.

Let G be a dynamic expression and $s = [G]_\approx$. The set of *all multisets of activities executable in s* is defined as $Exec(s) = \{\Upsilon \mid \exists H \in s \; \exists \widetilde{H} \; H \stackrel{\Upsilon}{\to} \widetilde{H}\}$. Here $H \stackrel{\Upsilon}{\to} \widetilde{H}$ is an inference by the rules from Table 3. It can be proved by induction on the structure of expressions that $\Upsilon \in Exec(s) \setminus \{\emptyset\}$ implies $\exists H \in s \; \Upsilon \in Now(H)$. The reverse statement does not hold, since the preconditions in the action rules disable executions of the activities with the lower-priority types from every $H \in s$, see [38,39].

The state s is *s-tangible (stochastically tangible)*, denoted by $stang(s)$, if $Exec(s) \subseteq \mathbb{N}_{fin}^{\mathcal{SL}}$. For an s-tangible state s we always have $\emptyset \in Exec(s)$ by rule **E**, hence, we may have $Exec(s) = \{\emptyset\}$. The state s is *w-tangible (waitingly tangible)*, denoted by $wtang(s)$, if $Exec(s) \subseteq \mathbb{N}_{fin}^{\mathcal{WL}} \setminus \{\emptyset\}$. The state s is *tangible*, denoted by $tang(s)$, if $stang(s)$ or $wtang(s)$, i.e. $Exec(s) \subseteq \mathbb{N}_{fin}^{\mathcal{SL}} \cup \mathbb{N}_{fin}^{\mathcal{WL}}$. Again, for a tangible state s we may have $\emptyset \in Exec(s)$ and $Exec(s) = \{\emptyset\}$. Otherwise, the state s is *vanishing*, denoted by $vanish(s)$, and in this case $Exec(s) \subseteq \mathbb{N}_{fin}^{\mathcal{IL}} \setminus \{\emptyset\}$.

If $\Upsilon \in Exec(s)$ and $\Upsilon \in \mathbb{N}_{fin}^{\mathcal{SL}} \cup \mathbb{N}_{fin}^{\mathcal{IL}}$ then by rules **P2s**, **P2i**, **Sy2s**, **Sy2i** and definition of $Exec(s)$ $\forall \Xi \subseteq \Upsilon$, $\Xi \neq \emptyset$, we have $\Xi \in Exec(s)$, i.e. $2^\Upsilon \setminus \{\emptyset\} \subseteq Exec(s)$.

Definition 3.10. 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 $\exists \Upsilon \; H \stackrel{\Upsilon}{\to} \widetilde{H}$ then $[\widetilde{H}]_\approx \in DR(G)$.

The set of *all s-tangible states from $DR(G)$* is denoted by $DR_{ST}(G)$, and the set of *all w-tangible states from $DR(G)$* is denoted by $DR_{WT}(G)$. The set of *all tangible states from $DR(G)$* is denoted by $DR_T(G) = DR_{ST}(G) \cup DR_{WT}(G)$. The set of *all vanishing states from $DR(G)$* is denoted by $DR_V(G)$. Then $DR(G) = DR_T(G) \uplus DR_V(G) = DR_{ST}(G) \uplus DR_{WT}(G) \uplus DR_V(G)$, where $\uplus$ denotes disjoint union.

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

Let $\Upsilon \in Exec(s) \setminus \{\emptyset\}$. The *probability that the multiset of stochastic multiactions Υ is ready for execution in s* or the *weight of the multiset of deterministic multiactions Υ which is ready for execution in s* is

$$PF(\Upsilon, s) = \begin{cases} \displaystyle\prod_{(\alpha,\rho)\in\Upsilon} \rho \cdot \prod_{\{\{(\beta,\chi)\}\in Exec(s) \mid (\beta,\chi)\notin\Upsilon\}} (1-\chi), & s \in DR_{ST}(G); \\ \displaystyle\sum_{(\alpha,\natural_l^\theta)\in\Upsilon} l, & s \in DR_{WT}(G) \cup DR_V(G). \end{cases}$$

In the case $\Upsilon = \emptyset$ and $s \in DR_{ST}(G)$ we define

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

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

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

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

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

Definition 3.11. 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 $L_G = \mathbb{N}_{fin}^{\mathcal{SDL}} \times (0; 1]$;
- the set of *transitions* is $\mathcal{T}_G = \{(s, (\Upsilon, PT(\Upsilon, s)), \tilde{s}) \mid s, \tilde{s} \in DR(G), \ \exists H \in s \ \exists \widetilde{H} \in \tilde{s} \ H \xrightarrow{\Upsilon} \widetilde{H}\}$;
- the *initial state* is $s_G = [G]_{\approx}$.

The transition system $TS(G)$ associated with a dynamic expression G describes all the steps (parallel executions) that occur at discrete time moments with some (one-step) probability and consist of multisets of activities. Every step consisting of stochastic (waiting, respectively) multiactions or the empty step (i.e. that consisting of the empty multiset of activities) occurs instantly after one discrete time unit delay. Each step consisting of immediate multiactions occurs instantly without any delay. The step can change the current state to a different 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 $(s, (\Upsilon, \mathcal{P}), \tilde{s}) \in \mathcal{T}_G$ will be written as $s \xrightarrow{\Upsilon}_{\mathcal{P}} \tilde{s}$. It is interpreted as follows: the probability to change from state s to $\tilde{s}$ as a result of executing Υ is $\mathcal{P}$.

From every s-tangible state the empty multiset of activities can always be executed by rule **E**. Hence, for s-tangible states, Υ may be the empty multiset, and its execution only decrements by one the timer values (if any) of the current state. Then we have a transition $s \xrightarrow{\emptyset}_{\mathcal{P}} \circlearrowleft s$ from an s-tangible state s to the tangible state $\circlearrowleft s = [\circlearrowleft H]_{\approx}$ for $H \in s \cap SatOpRegDynExpr$. Since structurally equivalent saturated operative dynamic expressions remain so after decreasing by one their timers, $\circlearrowleft s$ is unique for each s and the definition is correct. Thus, $\circlearrowleft s$ corresponds to applying the empty move rule to an arbitrary saturated operative dynamic expression from s, followed by taking the structural equivalence class of the result. We have to keep track of such executions, called the *empty moves*, since they affect the timers and have non-zero probabilities. This follows from the definition of $PF(\emptyset, s)$ and the fact that the probabilities of

stochastic multiactions belong to the interval $(0;1)$. When it holds $\circlearrowleft H = H$ for $H \in s \cap SatOpRegDynExpr$, we obtain $\circlearrowleft s = s$. Then the empty move from s is in the form of $s \xrightarrow{\emptyset}_{\mathcal{P}} s$, called the *empty loop*. For w-tangible and vanishing states Υ cannot be the empty multiset, since we must execute some immediate (waiting) multiactions from them at the current (next) moment.

The step probabilities belong to the interval $(0;1]$, being 1 if the only transition from an s-tangible state s is the empty move $s \xrightarrow{\emptyset}_1 \circlearrowleft s$, or if there is a single transition from a w-tangible or a vanishing state. We write $s \xrightarrow{\Upsilon} \tilde{s}$ if $\exists \mathcal{P}\ s \xrightarrow{\Upsilon}_{\mathcal{P}} \tilde{s}$ and $s \rightarrow \tilde{s}$ if $\exists \Upsilon\ s \xrightarrow{\Upsilon} \tilde{s}$.

Isomorphism is a coincidence of systems up to renaming of their components.

Definition 3.12. For dynamic expressions G and G', let $TS(G) = (S_G, L_G, \mathcal{T}_G, s_G)$ and $TS(G') = (S_{G'}, L_{G'}, \mathcal{T}_{G'}, s_{G'})$ be their transition systems. A mapping $\beta : S_G \rightarrow S_{G'}$ is an *isomorphism* between $TS(G)$ and $TS(G')$, denoted by $\beta : TS(G) \simeq TS(G')$, if

(1) β is a bijection such that $\beta(s_G) = s_{G'}$;
(2) $\forall s, \tilde{s} \in S_G\ \forall \Upsilon\ s \xrightarrow{\Upsilon}_{\mathcal{P}} \tilde{s} \ \Leftrightarrow\ \beta(s) \xrightarrow{\Upsilon}_{\mathcal{P}} \beta(\tilde{s})$.

Two transition systems $TS(G)$ and $TS(G')$ are *isomorphic*, denoted by $TS(G) \simeq TS(G')$, if $\exists \beta : TS(G) \simeq TS(G')$.

Definition 3.13. Two dynamic expressions G and G' are *equivalent with respect to transition systems*, denoted by $G =_{ts} G'$, if $TS(G) \simeq TS(G')$.

Let $N = (P_N, T_N, W_N, D_N, \Omega_N, \mathcal{L}_N, Q_N)$ be a LDTSDPN [38,39] and $Q, \widetilde{Q}$ be its states. Then the average sojourn time $SJ(Q)$, sojourn time variance $VAR(Q)$, probabilities $PM^*(Q, \widetilde{Q})$, transition relation $Q \twoheadrightarrow_{\mathcal{P}} \widetilde{Q}$, EDTMC $EDTMC(N)$, underlying SMC $SMC(N)$ and steady-state PMF for it are defined like the respective for dynamic expressions in [40,42]. Every marked and clocked plain dtsd-box [38,39] can be interpreted as an LDTSDPN. Thus, we can evaluate performance with the LDTSDPNs corresponding to dtsd-boxes and then transfer the results to the latter.

4. Stochastic equivalences

The semantic equivalence $=_{ts}$ is too discriminating, hence, we need weaker equivalence notions that 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 one such notion: 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 $[G]_{\approx}$ and $[G']_{\approx}$ of their transition systems should be related by a bisimulation having the following transfer property: if two states are related then in each of them the same multisets of multiactions can occur, leading with the identical overall probability from each of the

two states to *the same equivalence class* for every such multiset.

We follow the approaches of [4–6,48–51], 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 [48], in contrast to the reactive model, treated in [49], and we take transition probabilities instead of transition rates from [4–6,50,51]. Thus, step stochastic bisimulation equivalence that we define further is (in the probabilistic sense) comparable only with interleaving probabilistic bisimulation equivalence from [48], and our relation is obviously stronger.

In the definition below, we consider $\mathcal{L}(\Upsilon) \in \mathbb{N}^{\mathcal{L}}_{fin}$ for $\Upsilon \in \mathbb{N}^{\mathcal{SIL}}_{fin}$, i.e. (possibly empty) multisets of multiactions. The multiactions can be empty as well. In this case, $\mathcal{L}(\Upsilon)$ contains the elements $\emptyset$, but it is not empty itself.

Let G be a dynamic expression and $\mathcal{H} \subseteq DR(G)$. For any $s \in DR(G)$ and $A \in \mathbb{N}^{\mathcal{L}}_{fin}$, 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

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

We write $s \xrightarrow{A} \mathcal{H}$ if $\exists \mathcal{P}\ s \xrightarrow{A}_{\mathcal{P}} \mathcal{H}$. Further, we write $s \rightarrow_{\mathcal{P}} \mathcal{H}$ if $\exists A\ s \xrightarrow{A} \mathcal{H}$, 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

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

For $\tilde{s} \in DR(G)$, we write $s \xrightarrow{A}_{\mathcal{P}} \tilde{s}$ if $s \xrightarrow{A}_{\mathcal{P}} \{\tilde{s}\}$ and $s \xrightarrow{A} \tilde{s}$ if $\exists \mathcal{P}\ s \xrightarrow{A}_{\mathcal{P}} \tilde{s}$.

The mean value formula for the geometrical distribution allows us to calculate the average sojourn time in $s \in DR_T(G)$ as $SJ(s) = \frac{1}{1-PM(s,s)}$. The average sojourn time in each vanishing state $s \in DR_V(G)$ is $SJ(s) = 0$. Let $s \in DR(G)$.

The *average sojourn time in the state s* is

$$SJ(s) = \begin{cases} \frac{1}{1-PM(s,s)}, & s \in DR_T(G); \\ 0, & s \in DR_V(G). \end{cases}$$

The *average sojourn time vector* of G, denoted by SJ, has the elements $SJ(s)$, $s \in DR(G)$.

The *sojourn time variance in the state s* is

$$VAR(s) = \begin{cases} \frac{PM(s,s)}{(1-PM(s,s))^2}, & s \in DR_T(G); \\ 0, & s \in DR_V(G). \end{cases}$$

The *sojourn time variance vector* of G, denoted by VAR, has the elements $VAR(s)$, $s \in DR(G)$.

Definition 4.1. 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) $(s_1, s_2) \in \mathcal{R}$ implies $SJ(s_1) = 0 \;\Leftrightarrow\; SJ(s_2) = 0$ and
$\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}\; \forall A \in \mathbb{N}^{\mathcal{L}}_{fin}\; s_1 \stackrel{A}{\rightarrow}_{\mathcal{P}} \mathcal{H} \;\Leftrightarrow\; s_2 \stackrel{A}{\rightarrow}_{\mathcal{P}} \mathcal{H}$.

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

The condition $SJ(s_1) = 0 \;\Leftrightarrow\; SJ(s_2) = 0$ in item 2 of the definition above is needed to make difference between w-tangible states (all having at least one time unit sojourn times) and vanishing states (all having zero sojourn times). Both from w-tangible and vanishing states, no empty moves can be made, unlike s-tangible states, from which empty moves are always possible. When comparing dynamic expressions for step stochastic bisimulation equivalence, we can use empty moves only to make difference between s-tangible and other (w-tangible or vanishing) states.

We now define the multiaction transition systems, whose transitions are labeled with the multisets of multiactions, extracted from the corresponding activities.

Definition 4.2. Let G be a dynamic expression. The *(labeled probabilistic) multiaction transition system* of G is a quadruple $TS_{\mathcal{L}}(G) = (S_{\mathcal{L}}, L_{\mathcal{L}}, \mathcal{T}_{\mathcal{L}}, s_{\mathcal{L}})$, where

- $S_{\mathcal{L}} = DR(G)$;
- $L_{\mathcal{L}} = \mathbb{N}^{\mathcal{L}}_{fin} \times (0; 1]$;
- $\mathcal{T}_{\mathcal{L}} = \{(s, (A, PM_A(s, \{\tilde{s}\})), \tilde{s}) \mid s, \tilde{s} \in DR(G),\; s \stackrel{A}{\rightarrow} \tilde{s}\}$;
- $s_{\mathcal{L}} = [G]_{\approx}$.

The transition $(s, (A, \mathcal{P}), \tilde{s}) \in \mathcal{T}_{\mathcal{L}}$ will be written as $s \stackrel{A}{\rightarrow}_{\mathcal{P}} \tilde{s}$.

Let G and G' be dynamic expressions and $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$. Then the relation $\mathcal{R}$ can be interpreted as a step stochastic bisimulation between the transition systems $TS_{\mathcal{L}}(G)$ and $TS_{\mathcal{L}}(G')$, denoted by $\mathcal{R} : TS_{\mathcal{L}}(G) \underline{\leftrightarrow}_{ss} TS_{\mathcal{L}}(G')$, which is defined by analogy (excepting step semantics) with interleaving probabilistic bisimulation on generative probabilistic transition systems from [48].

The next proposition states that every step stochastic bisimulation binds s-tangible states only with s-tangible ones, and similarly for w-tangible and for vanishing states.

Proposition 4.3 ([39]). *Let G and G' be dynamic expressions and $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$. Then $\mathcal{R} \subseteq (DR_{ST}(G) \cup DR_{ST}(G'))^2 \uplus (DR_{WT}(G) \cup DR_{WT}(G'))^2 \uplus (DR_V(G) \cup DR_V(G'))^2$.*

Proposition 4.3 implies $\mathcal{R} \subseteq (DR_T(G) \cup DR_T(G'))^2 \uplus (DR_V(G) \cup DR_V(G'))^2$, since $DR_T(G) = DR_{ST}(G) \uplus DR_{WT}(G)$ and $DR_T(G') = DR_{ST}(G') \uplus DR_{WT}(G')$.

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 4.4 ([39]). *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'.*

The next theorem shows that both the semantics are bisimulation equivalent.

Theorem 4.5 ([39]). *For any static expression E, $TS(\overline{E}) \underline{\leftrightarrow}_{ss} RG(Box_{dtsd}(\overline{E}))$.*

We now compare the discrimination power of the stochastic equivalences.

Theorem 4.6 ([39]). *For dynamic expressions G and G' the* strict *implications hold:*

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

5. Reduction modulo equivalences

The equivalences which we proposed can be used to reduce transition systems and SMCs of expressions (reachability graphs and SMCs of dtsd-boxes). Reductions of graph-based models, like transition systems, reachability graphs and SMCs, result in those with less states (the graph nodes). The goal of the reduction is to decrease the number of states in the semantic representation of the modeled system while preserving its important qualitative and quantitative behavioral properties. Thus, the reduction allows one to simplify the functional and performance analysis. We now consider the quotient transition systems and Markov chains (SMCs, DTMCs, RDTMCs). Since the *quotient* sojourn time average and variance vectors, EDTMCs, SMCs, DTMCs and RDTMCs of expressions are defined by analogy with the respective *non-quotient* ones from [40,42], we do not present the latter definitions here, in order to avoid repetition.

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 $\forall \widetilde{\mathcal{K}} \in DR(G)/_{\mathcal{R}}\ \forall A \in \mathbb{N}^{\mathcal{L}}_{fin}\ s_1 \stackrel{A}{\rightarrow}_{\mathcal{P}} \widetilde{\mathcal{K}} \;\Leftrightarrow\; s_2 \stackrel{A}{\rightarrow}_{\mathcal{P}} \widetilde{\mathcal{K}}$. The previous equality is valid for all $s_1, s_2 \in \mathcal{K}$, hence, we can rewrite it as $\mathcal{K} \stackrel{A}{\rightarrow}_{\mathcal{P}} \widetilde{\mathcal{K}}$, 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 $\mathcal{K} \stackrel{A}{\rightarrow} \widetilde{\mathcal{K}}$ if $\exists \mathcal{P}\ \mathcal{K} \stackrel{A}{\rightarrow}_{\mathcal{P}} \widetilde{\mathcal{K}}$ and $\mathcal{K} \rightarrow \widetilde{\mathcal{K}}$ if $\exists A\ \mathcal{K} \stackrel{A}{\rightarrow} \widetilde{\mathcal{K}}$. The similar arguments allow us to write $\mathcal{K} \rightarrow_{\mathcal{P}} \widetilde{\mathcal{K}}$, where $\mathcal{P} = PM(\mathcal{K}, \widetilde{\mathcal{K}}) = PM(s_1, \widetilde{\mathcal{K}}) = PM(s_2, \widetilde{\mathcal{K}})$.

By the note after Proposition 4.3, $\mathcal{R} \subseteq (DR_T(G))^2 \uplus (DR_V(G))^2$. Hence, $\forall \mathcal{K} \in DR(G)/_{\mathcal{R}}$, all states from $\mathcal{K}$ are tangible, when $\mathcal{K} \in DR_T(G)/_{\mathcal{R}}$, or all of them are vanishing, when $\mathcal{K} \in DR_V(G)/_{\mathcal{R}}$.

The *average sojourn time in the equivalence class (with respect to $\mathcal{R}$) of states $\mathcal{K}$* is

$$SJ_{\mathcal{R}}(\mathcal{K}) = \begin{cases} \frac{1}{1-PM(\mathcal{K},\mathcal{K})}, & \mathcal{K} \in DR_T(G)/_{\mathcal{R}}; \\ 0, & \mathcal{K} \in DR_V(G)/_{\mathcal{R}}. \end{cases}$$

The *average sojourn time vector for the equivalence classes (with respect to $\mathcal{R}$) 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 (with respect to $\mathcal{R}$) of states $\mathcal{K}$* is

$$VAR_{\mathcal{R}}(\mathcal{K}) = \begin{cases} \frac{PM(\mathcal{K},\mathcal{K})}{(1-PM(\mathcal{K},\mathcal{K}))^2}, & \mathcal{K} \in DR_T(G)/_{\mathcal{R}}; \\ 0, & \mathcal{K} \in DR_V(G)/_{\mathcal{R}}. \end{cases}$$

The *sojourn time variance vector for the equivalence classes (with respect to $\mathcal{R}$) 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 4.4, $\mathcal{R}_{ss}(G)$ is the largest step stochastic autobisimulation on G. Based on the equivalence classes with respect to $\mathcal{R}_{ss}(G)$, the quotient (by $\underline{\leftrightarrow}_{ss}$) transition systems and quotient (by $\underline{\leftrightarrow}_{ss}$) underlying SMCs of expressions can be defined.

Those equivalence classes become the quotient states. The average sojourn time in a quotient state is that in the corresponding equivalence class. 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 5.1. Let G be a dynamic expression. The *quotient (by $\underline{\leftrightarrow}_{ss}$) (labeled probabilistic) transition system* of G is a quadruple $TS_{\underline{\leftrightarrow}_{ss}}(G) = (S_{\underline{\leftrightarrow}_{ss}}, L_{\underline{\leftrightarrow}_{ss}}, \mathcal{T}_{\underline{\leftrightarrow}_{ss}}, s_{\underline{\leftrightarrow}_{ss}})$:

- $S_{\underline{\leftrightarrow}_{ss}} = DR(G)/_{\mathcal{R}_{ss}(G)}$;
- $L_{\underline{\leftrightarrow}_{ss}} = \mathbb{N}^{\mathcal{L}}_{fin} \times (0;1]$;
- $\mathcal{T}_{\underline{\leftrightarrow}_{ss}} = \{(\mathcal{K}, (A, PM_A(\mathcal{K}, \widetilde{\mathcal{K}})), \widetilde{\mathcal{K}}) \mid \mathcal{K}, \widetilde{\mathcal{K}} \in DR(G)/_{\mathcal{R}_{ss}(G)},\ \mathcal{K} \stackrel{A}{\rightarrow} \widetilde{\mathcal{K}}\}$;
- $s_{\underline{\leftrightarrow}_{ss}} = [[G]_{\approx}]_{\mathcal{R}_{ss}(G)}$.

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

Let G be a dynamic expression. We define the relation $\mathcal{R}_{\mathcal{L}ss}(G) = \{(s, \mathcal{K}), (\mathcal{K}, s) \mid s \in \mathcal{K} \in DR(G)/_{\mathcal{R}_{ss}(G)}\}^+$, where $^+$ is the transitive closure operation. One can see that $\mathcal{R}_{\mathcal{L}ss}(G) \subseteq (DR(G) \cup DR(G)/_{\mathcal{R}_{ss}(G)})^2$ is an equivalence relation that partitions the set $DR(G) \cup DR(G)/_{\mathcal{R}_{ss}(G)}$ to the equivalence classes $\mathcal{L}_1, \ldots, \mathcal{L}_n$, defined as $\mathcal{L}_i = \mathcal{K}_i \cup \{\mathcal{K}_i\}$ $(1 \leq i \leq n)$, where $DR(G)/_{\mathcal{R}_{ss}(G)} = \{\mathcal{K}_1, \ldots, \mathcal{K}_n\}$. The relation $\mathcal{R}_{\mathcal{L}ss}(G)$ can be interpreted as a step stochastic bisimulation between the transition systems $TS_{\mathcal{L}}(G)$ and $TS_{\underline{\leftrightarrow}_{ss}}(G)$, denoted by $\mathcal{R}_{\mathcal{L}ss}(G) : TS_{\mathcal{L}}(G) \underline{\leftrightarrow}_{ss} TS_{\underline{\leftrightarrow}_{ss}}(G)$, which is defined by analogy (excepting step semantics) with interleaving probabilistic bisimulation on generative probabilistic transition systems from [48]. It is clear that from this viewpoint, $\mathcal{R}_{\mathcal{L}ss}(G)$ is also the union of all step stochastic bisimulations and largest step stochastic bisimulation between $TS_{\mathcal{L}}(G)$ and $TS_{\underline{\leftrightarrow}_{ss}}(G)$.

The quotient (by $\underline{\leftrightarrow}_{ss}$) reachability graphs are defined similarly to the quotient transition systems. Let $\simeq$ denote isomorphism between quotient transition systems and quotient reachability graphs that binds their initial states. The following proposition connects quotient (by $\underline{\leftrightarrow}_{ss}$) transition systems of the overlined static expressions and quotient reachability graphs of their dtsd-boxes.

Proposition 5.2 ([41]). *For any static expression E,*

$$TS_{\underline{\leftrightarrow}_{ss}}(\overline{E}) \simeq RG_{\underline{\leftrightarrow}_{ss}}(Box_{dtsd}(\overline{E})).$$

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 $VAR_{\underline{\leftrightarrow}_{ss}} = VAR_{\mathcal{R}_{ss}(G)}$.

Let G be a dynamic expression and $\mathcal{K}, \widetilde{\mathcal{K}} \in DR(G)/_{\mathcal{R}_{ss}(G)}$. The transition system $TS_{\underline{\leftrightarrow}_{ss}}(G)$ can have self-loops going from an equivalence class to itself which have a non-zero probability. The current equivalence class remains unchanged in this case.

Let $\mathcal{K} \rightarrow \mathcal{K}$. The *probability to stay in $\mathcal{K}$ due to k $(k \geq 1)$ self-loops* is $PM(\mathcal{K}, \mathcal{K})^k$.

Let $\mathcal{K} \rightarrow \widetilde{\mathcal{K}}$ and $\mathcal{K} \neq \widetilde{\mathcal{K}}$, i.e. $PM(\mathcal{K}, \mathcal{K}) < 1$. The *probability to move from $\mathcal{K}$ to $\widetilde{\mathcal{K}}$ by executing any multiset of activities after possible self-loops* is

$$PM^*(\mathcal{K}, \widetilde{\mathcal{K}}) = \left\{ \begin{array}{ll} PM(\mathcal{K}, \widetilde{\mathcal{K}}) \sum_{k=0}^{\infty} PM(\mathcal{K}, \mathcal{K})^k = \frac{PM(\mathcal{K}, \widetilde{\mathcal{K}})}{1 - PM(\mathcal{K}, \mathcal{K})}, & \mathcal{K} \rightarrow \mathcal{K}; \\ PM(\mathcal{K}, \widetilde{\mathcal{K}}), & \text{otherwise;} \end{array} \right\} =$$
$$SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) PM(\mathcal{K}, \widetilde{\mathcal{K}}).$$

Here $SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K})$ is the *quotient (by $\underline{\leftrightarrow}_{ss}$) self-loops abstraction factor in the equivalence class* $\mathcal{K}$. The *quotient (by $\underline{\leftrightarrow}_{ss}$) self-loops abstraction vector* of G, denoted by $SL_{\underline{\leftrightarrow}_{ss}}$, has the elements $SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K})$, $\mathcal{K} \in DR(G)/_{\mathcal{R}_{ss}(G)}$. The value $k = 0$ in the summation above corresponds to the case when no self-loops occur.

Let $\mathcal{K} \in DR_T(G)/_{\mathcal{R}_{ss}(G)}$. If there exist self-loops from $\mathcal{K}$ (i.e. if $\mathcal{K} \to \mathcal{K}$) then $PM(\mathcal{K},\mathcal{K}) > 0$ and $SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) = \frac{1}{1-PM(\mathcal{K},\mathcal{K})} = SJ_{\underline{\leftrightarrow}_{ss}}(\mathcal{K})$. Otherwise, if there exist no self-loops from $\mathcal{K}$ then $PM(\mathcal{K},\mathcal{K}) = 0$ and $SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) = 1 = \frac{1}{1-PM(\mathcal{K},\mathcal{K})} = SJ_{\underline{\leftrightarrow}_{ss}}(\mathcal{K})$. Thus, $\forall \mathcal{K} \in DR_T(G)/_{\mathcal{R}_{ss}(G)}$ $SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) = SJ_{\underline{\leftrightarrow}_{ss}}(\mathcal{K})$, hence, $\forall \mathcal{K} \in DR_T(G)/_{\mathcal{R}_{ss}(G)}$ with $PM(\mathcal{K},\mathcal{K}) < 1$ it holds $PM^*(\mathcal{K},\widetilde{\mathcal{K}}) = SJ_{\underline{\leftrightarrow}_{ss}}(\mathcal{K})PM(\mathcal{K},\widetilde{\mathcal{K}})$. Note that the self-loops from the equivalence classes of tangible states are of the empty or non-empty type, the latter produced by iteration, since empty loops are not possible from the equivalence classes of w-tangible states, but they are possible from the equivalence classes of s-tangible states, while non-empty loops are possible from the equivalence classes of both s-tangible and w-tangible states.

Let $\mathcal{K} \in DR_V(G)/_{\mathcal{R}_{ss}(G)}$. Then $\forall \mathcal{K} \in DR_V(G)/_{\mathcal{R}_{ss}(G)}$ $SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) \neq SJ_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) = 0$ and $\forall \mathcal{K} \in DR_V(G)/_{\mathcal{R}_{ss}(G)}$ with $PM(\mathcal{K},\mathcal{K}) < 1$it holds $PM^*(\mathcal{K},\widetilde{\mathcal{K}}) = SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K})PM(\mathcal{K},\widetilde{\mathcal{K}})$. If there exist self-loops from $\mathcal{K}$ then $PM^*(\mathcal{K},\widetilde{\mathcal{K}}) = \frac{PM(\mathcal{K},\widetilde{\mathcal{K}})}{1-PM(\mathcal{K},\mathcal{K})}$ when $PM(\mathcal{K},\mathcal{K}) < 1$. Otherwise, if there exist no self-loops from $\mathcal{K}$ then $PM^*(\mathcal{K},\widetilde{\mathcal{K}}) = PM(\mathcal{K},\widetilde{\mathcal{K}})$. Note that the self-loops from the equivalence classes of vanishing states are always of the non-empty type, produced by iteration, since empty loops are not possible from the equivalence classes of vanishing states.

Definition 5.3. Let G be a dynamic expression. The *quotient (by $\underline{\leftrightarrow}_{ss}$) EDTMC* of G, denoted by $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$, has the state space $DR(G)/_{\mathcal{R}_{ss}(G)}$, the initial state $[[G]_{\approx}]_{\mathcal{R}_{ss}(G)}$ and the transitions $\mathcal{K} \twoheadrightarrow_{\mathcal{P}} \widetilde{\mathcal{K}}$, if $\mathcal{K} \to \widetilde{\mathcal{K}}$ and $\mathcal{K} \neq \widetilde{\mathcal{K}}$, where $\mathcal{P} = PM^*(\mathcal{K},\widetilde{\mathcal{K}})$; or $\mathcal{K} \twoheadrightarrow_1 \mathcal{K}$, if $PM(\mathcal{K},\mathcal{K}) = 1$.

The *quotient (by $\underline{\leftrightarrow}_{ss}$) underlying SMC* of G, denoted by $SMC_{\underline{\leftrightarrow}_{ss}}(G)$, has the EDTMC $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ and the sojourn time in every $\mathcal{K} \in DR_T(G)/_{\mathcal{R}_{ss}(G)}$ is geometrically distributed with the parameter $1 - PM(\mathcal{K},\mathcal{K})$ while the sojourn time in every $\mathcal{K} \in DR_V(G)/_{\mathcal{R}_{ss}(G)}$ is equal to zero.

Let G be a dynamic expression. The elements $\mathcal{P}^*_{\underline{\leftrightarrow}_{ss}rs}$ $(1 \leq r,s \leq l = |DR(G)/_{\mathcal{R}_{ss}(G)}|)$ of the (one-step) transition probability matrix (TPM) $\mathbf{P}^*_{\underline{\leftrightarrow}_{ss}}$ for $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ are

$$\mathcal{P}^*_{\underline{\leftrightarrow}_{ss}rs} = \begin{cases} PM^*(\mathcal{K}_r,\mathcal{K}_s), & \mathcal{K}_r \to \mathcal{K}_s,\ r \neq s; \\ 1, & PM(\mathcal{K}_r,\mathcal{K}_r) = 1,\ r = s; \\ 0, & \text{otherwise.} \end{cases}$$

The transient (k-step, $k \in \mathbb{N}$) PMF $\psi^*_{\underline{\leftrightarrow}_{ss}}[k] = (\psi^*_{\underline{\leftrightarrow}_{ss}}[k](\mathcal{K}_1),\ldots,\psi^*_{\underline{\leftrightarrow}_{ss}}[k](\mathcal{K}_l))$ for $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ is calculated as $\psi^*_{\underline{\leftrightarrow}_{ss}}[k] = \psi^*_{\underline{\leftrightarrow}_{ss}}[0](\mathbf{P}^*_{\underline{\leftrightarrow}_{ss}})^k$, where $\psi^*_{\underline{\leftrightarrow}_{ss}}[0] = (\psi^*_{\underline{\leftrightarrow}_{ss}}[0](\mathcal{K}_1),\ldots,\psi^*_{\underline{\leftrightarrow}_{ss}}[0](\mathcal{K}_l))$ is the initial PMF: $\psi^*_{\underline{\leftrightarrow}_{ss}}[0](\mathcal{K}_r) = \begin{cases} 1, & \mathcal{K}_r = [[G]_{\approx}]_{\mathcal{R}_{ss}(G)}; \\ 0, & \text{otherwise.} \end{cases}$
Note that $\psi^*_{\underline{\leftrightarrow}_{ss}}[k+1] = \psi^*_{\underline{\leftrightarrow}_{ss}}[k]\mathbf{P}^*_{\underline{\leftrightarrow}_{ss}}$ $(k \in \mathbb{N})$.

The steady-state PMF $\psi^*_{\underline{\leftrightarrow}_{ss}} = (\psi^*_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_1),\ldots,\psi^*_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_l))$ for $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ is a

solution of the equation system

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

where $\mathbf{I}$ is the identity matrix of order l and $\mathbf{0}$ ($\mathbf{1}$) is a row vector of l values 0 (1).

Note that the vector $\psi^*_{\underline{\leftrightarrow}_{ss}}$ exists and is unique if $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ is ergodic. Then $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ has a single steady state, and we have $\psi^*_{\underline{\leftrightarrow}_{ss}} = \lim_{k\to\infty}\psi^*_{\underline{\leftrightarrow}_{ss}}[k]$.

The steady-state PMF for the quotient (by $\underline{\leftrightarrow}_{ss}$) underlying semi-Markov chain $SMC_{\underline{\leftrightarrow}_{ss}}(G)$ is calculated via multiplying every $\psi^*_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_r)$ $(1 \le r \le l)$ by the average sojourn time $SJ_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_r)$ in the equivalence class $\mathcal{K}_r$, followed by normalizing the resulting values. Note that for each equivalence class $\mathcal{K} \in DR_T(G)/_{\mathcal{R}_{ss}(G)}$ we have $SJ_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) \ge 1$ and for each equivalence class $\mathcal{K} \in DR_V(G)/_{\mathcal{R}_{ss}(G)}$ we have $SJ_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) = 0$.

Thus, the steady-state PMF $\varphi_{\underline{\leftrightarrow}_{ss}} = (\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_1), \ldots, \varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_l))$ for $SMC_{\underline{\leftrightarrow}_{ss}}(G)$ is

$$\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_r) = \begin{cases} \frac{\psi^*_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_r) SJ_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_r)}{\sum_{s=1}^{l} \psi^*_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_s) SJ_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_s)}, & \mathcal{K}_r \in DR_T(G)/_{\mathcal{R}_{ss}(G)}; \\ 0, & \mathcal{K}_r \in DR_V(G)/_{\mathcal{R}_{ss}(G)}. \end{cases}$$

Let $\simeq$ denote isomorphism between SMCs that binds their initial states, i.e. their EDTMCs are isomorphic and the sojourn times in the isomorphic states are identically distributed. The following proposition establishes a connection between quotient (by $\underline{\leftrightarrow}_{ss}$) SMCs of the overlined static expressions and quotient SMCs of their dtsd-boxes.

Proposition 5.4 ([41])**.** *For any static expression E*

$$SMC_{\underline{\leftrightarrow}_{ss}}(\overline{E}) \simeq SMC_{\underline{\leftrightarrow}_{ss}}(Box_{dtsd}(\overline{E})).$$

The quotients of both transition systems and underlying SMCs are the minimal reductions of the mentioned objects modulo step stochastic bisimulations. The quotients can be used to simplify analysis of system properties which are preserved by $\underline{\leftrightarrow}_{ss}$, since potentially less states should be examined for it.

Consider quotient (by $\underline{\leftrightarrow}_{ss}$) DTMCs with the state change probabilities $PM(\mathcal{K}, \widetilde{\mathcal{K}})$.

Definition 5.5. Let G be a dynamic expression. The *quotient (by $\underline{\leftrightarrow}_{ss}$) 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 $\mathcal{K} \to_{\mathcal{P}} \widetilde{\mathcal{K}}$, where $\mathcal{P} = PM(\mathcal{K}, \widetilde{\mathcal{K}})$.

The steady-state PMF $\psi_{\underline{\leftrightarrow}_{ss}}$ for $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$ is defined like the respective $\psi^*_{\underline{\leftrightarrow}_{ss}}$ for $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ and is related with the steady-state PMF $\varphi_{\underline{\leftrightarrow}_{ss}}$ for $SMC_{\underline{\leftrightarrow}_{ss}}(G)$.

Proposition 5.6. *Let G be a dynamic expression, $\varphi_{\underline{\leftrightarrow}_{ss}}$ be the steady-state PMF for $SMC_{\underline{\leftrightarrow}_{ss}}(G)$ and $\psi_{\underline{\leftrightarrow}_{ss}}$ be the steady-state PMF for $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$. Then $\forall \mathcal{K} \in DR(G)/_{\mathcal{R}_{ss}(G)}$*

$$\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) = \begin{cases} \frac{\psi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K})}{\sum_{\widetilde{\mathcal{K}} \in DR_T(G)/_{\mathcal{R}_{ss}(G)}} \psi_{\underline{\leftrightarrow}_{ss}}(\widetilde{\mathcal{K}})}, & \mathcal{K} \in DR_T(G)/_{\mathcal{R}_{ss}(G)}; \\ 0, & \mathcal{K} \in DR_V(G)/_{\mathcal{R}_{ss}(G)}. \end{cases}$$

Proof. Like that of the corresponding 'non-quotient' Proposition 4 from [40]. □

Eliminating equivalence classes (with respect to $\mathcal{R}_{ss}(G)$) of vanishing states from the quotient (by $\underline{\leftrightarrow}_{ss}$) DTMCs of expressions results in the reductions of the DTMCs.

Definition 5.7. Let G be a dynamic expression. The *reduced quotient (by $\underline{\leftrightarrow}_{ss}$) DTMC* of G, denoted by $RDTMC_{\underline{\leftrightarrow}_{ss}}(G)$, is defined like $RDTMC(G)$ in [40,42], but it is constructed from $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$ instead of $DTMC(G)$.

The steady-state PMF $\psi^{\diamond}_{\underline{\leftrightarrow}_{ss}}$ for $RDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ is defined like the respective $\psi^{*}_{\underline{\leftrightarrow}_{ss}}$ for $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ and is related with the steady-state PMF $\varphi_{\underline{\leftrightarrow}_{ss}}$ for $SMC_{\underline{\leftrightarrow}_{ss}}(G)$.

Proposition 5.8. *Let G be a dynamic expression, $\varphi_{\underline{\leftrightarrow}_{ss}}$ be the steady-state PMF for $SMC_{\underline{\leftrightarrow}_{ss}}(G)$ and $\psi^{\diamond}_{\underline{\leftrightarrow}_{ss}}$ be the steady-state PMF for $RDTMC_{\underline{\leftrightarrow}_{ss}}(G)$. Then $\forall \mathcal{K} \in DR(G)/_{\mathcal{R}_{ss}(G)}$*

$$\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) = \begin{cases} \psi^{\diamond}_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}), & \mathcal{K} \in DR_T(G)/_{\mathcal{R}_{ss}(G)}; \\ 0, & \mathcal{K} \in DR_V(G)/_{\mathcal{R}_{ss}(G)}. \end{cases}$$

Proof. Like that of the corresponding 'non-quotient' Proposition 5 from [40]. □

6. Stationary behavior

Let us examine how the proposed equivalences can be used to compare the behavior 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 behavior, since there can exist a deadlock (blocking) within the body (the second argument) of iteration when the corresponding subprocess does not reach its final state by some reasons. In particular, consider the expression $\mathsf{Stop} = (\{g\}, \frac{1}{2})$ rs g specifying the non-terminating process that performs only empty loops with probability 1. 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.

We consider only the process expressions such that their underlying SMCs contain exactly one closed communication class of states, and this class should be ergodic to ensure uniqueness of the stationary distribution, which is also the limiting one. The states not belonging to that class do not disturb the uniqueness, since the closed communication class is single, hence, they all are transient. Then, for each transient state, the steady-state probability to be in it is zero while the steady-state probability to enter into the ergodic class starting from that state is equal to one.

6.1. Steady state, residence time 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. There-

fore, the mean recurrence time for an equivalence class is the same for both expressions.

Proposition 6.1. *Let G, G' be dynamic expressions with $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$ and φ be the steady-state PMF for $SMC(G)$, φ' be the steady-state PMF for $SMC(G')$. Then $\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$*

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

Proof. Like that of Proposition 6 from [37] or that of Proposition 10 from [41]. □

Let G be a dynamic expression and φ be the steady-state PMF for $SMC(G)$, $\varphi_{\underline{\leftrightarrow}_{ss}}$ be the steady-state PMF for $SMC_{\underline{\leftrightarrow}_{ss}}(G)$. By Proposition 6.1 (modified for $\mathcal{R}_{\mathcal{L}ss}(G)$), we have $\forall \mathcal{K} \in DR(G)/_{\mathcal{R}_{ss}(G)}$

$$\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) = \sum_{s \in \mathcal{K}} \varphi(s).$$

Thus, for every equivalence class $\mathcal{K} \in DR(G)/_{\mathcal{R}_{ss}(G)}$, the value of $\varphi_{\underline{\leftrightarrow}_{ss}}$ for $\mathcal{K}$ is the sum of all values of φ corresponding to the states from $\mathcal{K}$.

By Proposition 6.1, $\underline{\leftrightarrow}_{ss}$ preserves the quantitative properties of the stationary behavior (the level of SMCs). We now intend to demonstrate that the qualitative properties of the stationary behavior based on the multiaction labels are preserved as well (the level of transition systems).

Definition 6.2. A *derived step trace* of a dynamic expression G is a chain $\Sigma = A_1 \cdots A_n \in (\mathbb{N}^{\mathcal{L}}_{fin})^*$, where $\exists s \in DR(G)\ s \xrightarrow{\Upsilon_1} s_1 \xrightarrow{\Upsilon_2} \cdots \xrightarrow{\Upsilon_n} s_n,\ \mathcal{L}(\Upsilon_i) = A_i\ (1 \leq i \leq n)$. Then the *probability to execute the derived step trace Σ in s* is

$$PT(\Sigma, s) = \sum_{\{\Upsilon_1, \ldots, \Upsilon_n \mid s = s_0 \xrightarrow{\Upsilon_1} s_1 \xrightarrow{\Upsilon_2} \cdots \xrightarrow{\Upsilon_n} s_n,\ \mathcal{L}(\Upsilon_i) = A_i\ (1 \leq i \leq n)\}} \prod_{i=1}^{n} PT(\Upsilon_i, s_{i-1}).$$

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 6.3. *Let G, G' be dynamic expressions with $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$ and φ be the steady-state PMF for $SMC(G)$, φ' be the steady-state PMF for $SMC(G')$ and Σ be a derived step trace of G and G'. Then $\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$*

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

Proof. Like that of Theorem 4 from [37]. □

Let G be a dynamic expression, φ be the steady-state PMF for $SMC(G)$, $\varphi_{\underline{\leftrightarrow}_{ss}}$ be the steady-state PMF for $SMC_{\underline{\leftrightarrow}_{ss}}(G)$ and Σ be a derived step trace of G. By

Theorem 6.3 (modified for $\mathcal{R}_{\mathcal{L}ss}(G)$), we get $\forall \mathcal{K} \in DR(G)/_{\mathcal{R}_{ss}(G)}$

$$\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K})PT(\Sigma,\mathcal{K}) = \sum_{s\in\mathcal{K}} \varphi(s)PT(\Sigma,s),$$

where $\forall s \in \mathcal{K}\ PT(\Sigma,\mathcal{K}) = PT(\Sigma,s)$.

The following proposition demonstrates that, for two dynamic expressions related by $\underline{\leftrightarrow}_{ss}$, the sojourn time averages in an equivalence class coincide, as well as the sojourn time variances in it.

Proposition 6.4. *Let G, G' be dynamic expressions with $\mathcal{R} : G\underline{\leftrightarrow}_{ss}G'$. Then $\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$*

$$SJ_{\mathcal{R}\cap(DR(G))^2}(\mathcal{H} \cap DR(G)) = SJ_{\mathcal{R}\cap(DR(G'))^2}(\mathcal{H} \cap DR(G')),$$
$$VAR_{\mathcal{R}\cap(DR(G))^2}(\mathcal{H} \cap DR(G)) = VAR_{\mathcal{R}\cap(DR(G'))^2}(\mathcal{H} \cap DR(G')).$$

Proof. Like that of Proposition 7 from [37]. □

6.2. 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. Some of the indices are calculated using the average or the variance of sojourn time in a set of similar states. The similarity of states is captured by an equivalence relation, hence, the sets are the equivalence classes. Proposition 6.1, Theorem 6.3 and Proposition 6.4 guarantee 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 dtsdPBC.

Next, it is easier to evaluate performance using an SMC with less states, since in this case the size of the transition probability matrix will be smaller, and we shall 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 dtsdPBC.
(2) The transition system of the expression is constructed.
(3) The step stochastic autobisimulation equivalence for the expression is defined.
(4) The quotient underlying SMC is derived from the quotient transition system.
(5) Stationary probabilities and performance indices are obtained from the SMC.

The limitation of the method above is its applicability only to the expressions such that their underlying SMCs contain exactly one closed communication class of states, and this class should also be ergodic to ensure uniqueness of the stationary distribution. If an SMC contains several closed communication classes of states that are all ergodic then several stationary distributions may exist, which depend on the initial PMF. There is an analytical method to determine stationary probabilities for SMCs of this kind as well [52]. The underlying SMC of every process expression has only one initial PMF (that at the time moment 0), hence, the stationary distribution will be unique in this case too. The general steady-state probabilities are then calculated as the sum of the stationary probabilities of all the ergodic classes of states, weighted by the

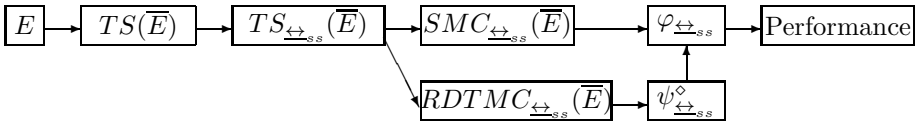


Figure 1. Equivalence-based simplification of performance evaluation.

probabilities to enter into these classes, starting from the initial state and passing through some transient states. In addition, it is worth applying the method only to the systems with similar subprocesses.

Alternatively, the results at the end of Section 5 allow us to simplify the steps 4 and 5 of the method above by constructing the reduced quotient DTMC from the quotient transition system, followed by calculating the stationary probabilities of the quotient underlying SMC using that DTMC, and then obtaining the performance indices. In more detail, the quotient transition system $TS_{\underline{\leftrightarrow}_{ss}}(\overline{E})$ provides the information both about the probabilities to move between the equivalence classes of states $PM(\mathcal{K}, \widetilde{\mathcal{K}})$ and about the equivalence classes of vanishing states $DR_V(\overline{E})/_{\mathcal{R}_{ss}(\overline{E})}$. That information is used to construct the reordered quotient transition probability matrix (TPM) $\mathbf{P}_{r\underline{\leftrightarrow}_{ss}}$, from which the TPM $\mathbf{P}^{\diamond}_{\underline{\leftrightarrow}_{ss}}$ for $RDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{E})$ is further obtained.

Figure 1 presents the main stages of the standard and alternative equivalence-based simplification of performance evaluation described above.

7. Generalized shared memory system with maintenance

Consider a model of two processors accessing a common shared memory described in [23,26,46] in the continuous time setting on GSPNs. We shall analyze this shared memory system in the discrete time setting of dtsdPBC, where concurrent execution of activities is possible, while no two transitions of a GSPN may fire simultaneously (in parallel). We also add to the system a feature of the memory maintenance. Our *generalized* model parameterizes the *standard* shared memory system by treating the probabilities and weights from its specification as variables (parameters). The model behaves as follows. After activation of the system (turning the computer on), two processors are active, and the common memory is available. Each processor can request an access to the memory after which the instantaneous decision is made, if the memory is available. When the decision is made in favour of a processor, it starts acquisition of the memory and the other processor should wait until the former one ends its memory operations, and the system returns to the state with both active processors and available common memory. If the memory is available and not required then its maintenance can be initiated, followed by the short memory service works (for example, the checksum test) during a fixed period of time, after which the memory becomes available again. If the memory requirement and its maintenance initiation happen at the same time then the service works start and no decision on the memory allocation is made while the memory is maintained. The diagram of the system is depicted in Figure 2.

7.1. The concrete system

The meaning of actions from the expressions which will specify the system modules is as follows. The action a corresponds to the system activation. The action c specifies the

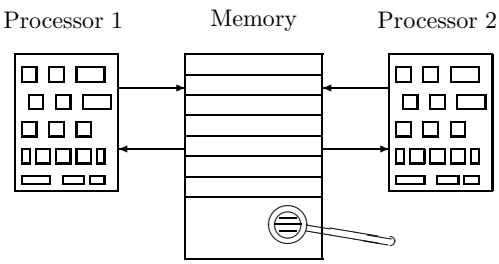


Figure 2. The diagram of the shared memory system with maintenance.

memory maintenance initiation. The action e means the short memory service (taking a fixed time of 1 unit). The actions r_i $(1 \le i \le 2)$ represent the common memory request (whose probability is 10 times greater than that of the maintenance initiation) of processor i. The actions d_i correspond to the (instantaneous) decision on the memory allocation in favour of the processor i. The actions m_i represent the common memory access of processor i. The other actions are used for communication purposes only via synchronization, and we abstract from them later using restriction. For $a_1, \ldots, a_n \in Act$ $(n \in \mathbb{N})$, we abbreviate $\mathsf{sy}\ a_1 \cdots \mathsf{sy}\ a_n\ \mathsf{rs}\ a_1 \cdots \mathsf{rs}\ a_n$ to $\mathsf{sr}\ (a_1, \ldots, a_n)$.

We take general values for all specified multiaction probabilities and weights. Let all stochastic multiactions have the same generalized probability $\rho \in (0; 1)$ and all deterministic ones have the same generalized weight $l \in \mathbb{R}_{>0}$. The specification K of the generalized shared memory system with maintenance follows.

The static expression of the first processor is

$$K_1 = [(\{x_1\}, \rho) * ((\{r_1\}, \rho); (\{d_1, y_1\}, \natural_l^0); (\{m_1, z_1\}, \rho)) * \mathsf{Stop}].$$

The static expression of the second processor is

$$K_2 = [(\{x_2\}, \rho) * ((\{r_2\}, \rho); (\{d_2, y_2\}, \natural_l^0); (\{m_2, z_2\}, \rho)) * \mathsf{Stop}].$$

The static expression of the shared memory is

$$K_3 = [(\{a, \widehat{x_1}, \widehat{x_2}\}, \rho) * ((((\{c\}, \tfrac{\rho}{10}); (\{e\}, \natural_l^1))[]((\{\widehat{y_1}\}, \natural_l^0); (\{\widehat{z_1}\}, \rho))[] \\ ((\{\widehat{y_2}\}, \natural_l^0); (\{\widehat{z_2}\}, \rho))) * \mathsf{Stop}].$$

The static expression of the generalized shared memory system with maintenance is

$$K = (K_1 \| K_2 \| K_3)\ \mathsf{sr}\ (x_1, x_2, y_1, y_2, z_1, z_2).$$

Let us illustrate an effect of synchronization. As a result of the synchronization of immediate multiactions $(\{d_i, y_i\}, \natural_l^0)$ and $(\{\widehat{y_i}\}, \natural_l^0)$ we get is $(\{d_i\}, \natural_{2l})$ $(1 \le i \le 2)$. The synchronization of stochastic multiactions $(\{m_i, z_i\}, \rho)$ and $(\{\widehat{z_i}\}, \rho)$ produces $(\{m_i\}, \rho^2)$ $(1 \le i \le 2)$. The result of synchronization of $(\{a, \widehat{x_1}, \widehat{x_2}\}, \rho)$ with $(\{x_1\}, \rho)$ is $(\{a, \widehat{x_2}\}, \rho^2)$, and that of synchronization of $(\{a, \widehat{x_1}, \widehat{x_2}\}, \rho)$ with $(\{x_2\}, \rho)$ is $(\{a, \widehat{x_1}\}, \rho^2)$. After applying synchronization to $(\{a, \widehat{x_2}\}, \rho^2)$ and $(\{x_2\}, \rho)$, as well as to $(\{a, \widehat{x_1}\}, \rho^2)$ and $(\{x_1\}, \rho)$, we get the same activity $(\{a\}, \rho^3)$.

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

$$\tilde{s}_1 = [([\overline{(\{x_1\},\rho)} * ((\{r_1\},\rho);(\{d_1,y_1\},\natural_l^0);(\{m_1,z_1\},\rho)) * \mathsf{Stop}]\|$$
$$\overline{[(\{x_2\},\rho)} * ((\{r_2\},\rho);(\{d_2,y_2\},\natural_l^0);(\{m_2,z_2\},\rho)) * \mathsf{Stop}]\|$$
$$\overline{[(\{a,\widehat{x_1},\widehat{x_2}\},\rho)} * ((((\{c\},\tfrac{\rho}{10});(\{e\},\natural_l^1))[](((\{\widehat{y_1}\},\natural_l^0);(\{\widehat{z_1}\},\rho))[]$$
$$(((\{\widehat{y_2}\},\natural_l^0);(\{\widehat{z_2}\},\rho))) * \mathsf{Stop}])\ \mathsf{sr}\ (x_1,x_2,y_1,y_2,z_1,z_2)]_{\approx},$$

$$\tilde{s}_2 = [([(\{x_1\},\rho) * \overline{((\{r_1\},\rho)};(\{d_1,y_1\},\natural_l^0);(\{m_1,z_1\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\rho) * \overline{((\{r_2\},\rho)};(\{d_2,y_2\},\natural_l^0);(\{m_2,z_2\},\rho)) * \mathsf{Stop}]\|$$
$$\overline{[(\{a,\widehat{x_1},\widehat{x_2}\},\rho) * ((((\{c\},\tfrac{\rho}{10});(\{e\},\natural_l^1))[](((\{\widehat{y_1}\},\natural_l^0);(\{\widehat{z_1}\},\rho))[]}$$
$$\overline{(((\{\widehat{y_2}\},\natural_l^0);(\{\widehat{z_2}\},\rho)))} * \mathsf{Stop}])\ \mathsf{sr}\ (x_1,x_2,y_1,y_2,z_1,z_2)]_{\approx},$$

$$\tilde{s}_3 = [([(\{x_1\},\rho) * \overline{((\{r_1\},\rho)};(\{d_1,y_1\},\natural_l^0);(\{m_1,z_1\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\rho) * (\overline{(\{r_2\},\rho)};(\{d_2,y_2\},\natural_l^0);(\{m_2,z_2\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2}\},\rho) * ((((\{c\},\tfrac{\rho}{10});(\{e\},\natural_l^1)^1)[](((\{\widehat{y_1}\},\natural_l^0);(\{\widehat{z_1}\},\rho))[]$$
$$(((\{\widehat{y_2}\},\natural_l^0);(\{\widehat{z_2}\},\rho))) * \mathsf{Stop}])\ \mathsf{sr}\ (x_1,x_2,y_1,y_2,z_1,z_2)]_{\approx},$$

$$\tilde{s}_4 = [([(\{x_1\},\rho) * ((\{r_1\},\rho);\overline{(\{d_1,y_1\},\natural_l^0)};(\{m_1,z_1\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\rho) * (\overline{(\{r_2\},\rho)};(\{d_2,y_2\},\natural_l^0);(\{m_2,z_2\},\rho)) * \mathsf{Stop}]\|$$
$$\overline{[(\{a,\widehat{x_1},\widehat{x_2}\},\rho) * ((((\{c\},\tfrac{\rho}{10});(\{e\},\natural_l^1))[](((\{\widehat{y_1}\},\natural_l^0);(\{\widehat{z_1}\},\rho))[]}$$
$$\overline{(((\{\widehat{y_2}\},\natural_l^0);(\{\widehat{z_2}\},\rho)))} * \mathsf{Stop}])\ \mathsf{sr}\ (x_1,x_2,y_1,y_2,z_1,z_2)]_{\approx},$$

$$\tilde{s}_5 = [([(\{x_1\},\rho) * (\overline{(\{r_1\},\rho)};(\{d_1,y_1\},\natural_l^0);(\{m_1,z_1\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\rho) * ((\{r_2\},\rho);\overline{(\{d_2,y_2\},\natural_l^0)};(\{m_2,z_2\},\rho)) * \mathsf{Stop}]\|$$
$$\overline{[(\{a,\widehat{x_1},\widehat{x_2}\},\rho) * ((((\{c\},\tfrac{\rho}{10});(\{e\},\natural_l^1))[](((\{\widehat{y_1}\},\natural_l^0);(\{\widehat{z_1}\},\rho))[]}$$
$$\overline{(((\{\widehat{y_2}\},\natural_l^0);(\{\widehat{z_2}\},\rho)))} * \mathsf{Stop}])\ \mathsf{sr}\ (x_1,x_2,y_1,y_2,z_1,z_2)]_{\approx},$$

$$\tilde{s}_6 = [([(\{x_1\},\rho) * ((\{r_1\},\rho);\overline{(\{d_1,y_1\},\natural_l^0)};(\{m_1,z_1\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\rho) * ((\{r_2\},\rho);\overline{(\{d_2,y_2\},\natural_l^0)};(\{m_2,z_2\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2}\},\rho) * ((((\{c\},\tfrac{\rho}{10});\overline{(\{e\},\natural_l^1)^1})[](((\{\widehat{y_1}\},\natural_l^0);(\{\widehat{z_1}\},\rho))[]$$
$$(((\{\widehat{y_2}\},\natural_l^0);(\{\widehat{z_2}\},\rho))) * \mathsf{Stop}])\ \mathsf{sr}\ (x_1,x_2,y_1,y_2,z_1,z_2)]_{\approx},$$

$$\tilde{s}_7 = [([(\{x_1\},\rho) * ((\{r_1\},\rho);\overline{(\{d_1,y_1\},\natural_l^0)};(\{m_1,z_1\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\rho) * (\overline{(\{r_2\},\rho)};(\{d_2,y_2\},\natural_l^0);(\{m_2,z_2\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2}\},\rho) * ((((\{c\},\tfrac{\rho}{10});\overline{(\{e\},\natural_l^1)^1})[](((\{\widehat{y_1}\},\natural_l^0);(\{\widehat{z_1}\},\rho))[]$$
$$(((\{\widehat{y_2}\},\natural_l^0);(\{\widehat{z_2}\},\rho))) * \mathsf{Stop}])\ \mathsf{sr}\ (x_1,x_2,y_1,y_2,z_1,z_2)]_{\approx},$$

$$\tilde{s}_8 = [([(\{x_1\},\rho) * (\overline{(\{r_1\},\rho)}; (\{d_1,y_1\},\natural_l^0); (\{m_1,z_1\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\rho) * ((\{r_2\},\rho); \overline{(\{d_2,y_2\},\natural_l^0)}; (\{m_2,z_2\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2}\},\rho) * ((((\{c\},\tfrac{\rho}{10}); \overline{(\{e\},\natural_l^1)^1})[]((\{\widehat{y_1}\},\natural_l^0); (\{\widehat{z_1}\},\rho))[]$$
$$((\{\widehat{y_2}\},\natural_l^0); (\{\widehat{z_2}\},\rho))) * \mathsf{Stop}])\ \mathsf{sr}\ (x_1,x_2,y_1,y_2,z_1,z_2)]_{\approx},$$

$$\tilde{s}_9 = [([(\{x_1\},\rho) * ((\{r_1\},\rho); \overline{(\{d_1,y_1\},\natural_l^0)}; (\{m_1,z_1\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\rho) * ((\{r_2\},\rho); \overline{(\{d_2,y_2\},\natural_l^0)}; (\{m_2,z_2\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2}\},\rho) * \overline{((((\{c\},\tfrac{\rho}{10}); (\{e\},\natural_l^1))[]((\{\widehat{y_1}\},\natural_l^0); (\{\widehat{z_1}\},\rho))[]}$$
$$\overline{((\{\widehat{y_2}\},\natural_l^0); (\{\widehat{z_2}\},\rho)))} * \mathsf{Stop}])\ \mathsf{sr}\ (x_1,x_2,y_1,y_2,z_1,z_2)]_{\approx},$$

$$\tilde{s}_{10} = [([(\{x_1\},\rho) * ((\{r_1\},\rho); (\{d_1,y_1\},\natural_l^0); \overline{(\{m_1,z_1\},\rho)}) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\rho) * (\overline{(\{r_2\},\rho)}; (\{d_2,y_2\},\natural_l^0); (\{m_2,z_2\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2}\},\rho) * ((((\{c\},\tfrac{\rho}{10}); (\{e\},\natural_l^1))[]((\{\widehat{y_1}\},\natural_l^0); \overline{(\{\widehat{z_1}\},\rho)})[]$$
$$((\{\widehat{y_2}\},\natural_l^0); (\{\widehat{z_2}\},\rho))) * \mathsf{Stop}])\ \mathsf{sr}\ (x_1,x_2,y_1,y_2,z_1,z_2)]_{\approx},$$

$$\tilde{s}_{11} = [([(\{x_1\},\rho) * (\overline{(\{r_1\},\rho)}; (\{d_1,y_1\},\natural_l^0); (\{m_1,z_1\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\rho) * ((\{r_2\},\rho); (\{d_2,y_2\},\natural_l^0); \overline{(\{m_2,z_2\},\rho)}) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2}\},\rho) * ((((\{c\},\tfrac{\rho}{10}); (\{e\},\natural_l^1))[]((\{\widehat{y_1}\},\natural_l^0); (\{\widehat{z_1}\},\rho))[]$$
$$((\{\widehat{y_2}\},\natural_l^0); \overline{(\{\widehat{z_2}\},\rho)})) * \mathsf{Stop}])\ \mathsf{sr}\ (x_1,x_2,y_1,y_2,z_1,z_2)]_{\approx},$$

$$\tilde{s}_{12} = [([(\{x_1\},\rho) * ((\{r_1\},\rho); \overline{(\{d_1,y_1\},\natural_l^0)}; \overline{(\{m_1,z_1\},\rho)}) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\rho) * ((\{r_2\},\rho); \overline{(\{d_2,y_2\},\natural_l^0)}; (\{m_2,z_2\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2}\},\rho) * ((((\{c\},\tfrac{\rho}{10}); (\{e\},\natural_l^1))[]((\{\widehat{y_1}\},\natural_l^0); \overline{(\{\widehat{z_1}\},\rho)})[]$$
$$((\{\widehat{y_2}\},\natural_l^0); (\{\widehat{z_2}\},\rho))) * \mathsf{Stop}])\ \mathsf{sr}\ (x_1,x_2,y_1,y_2,z_1,z_2)]_{\approx},$$

$$\tilde{s}_{13} = [([(\{x_1\},\rho) * ((\{r_1\},\rho); \overline{(\{d_1,y_1\},\natural_l^0)}; (\{m_1,z_1\},\rho)) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\rho) * ((\{r_2\},\rho); (\{d_2,y_2\},\natural_l^0); \overline{(\{m_2,z_2\},\rho)}) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2}\},\rho) * ((((\{c\},\tfrac{\rho}{10}); (\{e\},\natural_l^1))[]((\{\widehat{y_1}\},\natural_l^0); (\{\widehat{z_1}\},\rho))[]$$
$$((\{\widehat{y_2}\},\natural_l^0); \overline{(\{\widehat{z_2}\},\rho)})) * \mathsf{Stop}])\ \mathsf{sr}\ (x_1,x_2,y_1,y_2,z_1,z_2)]_{\approx},$$

We have $DR_{ST}(\overline{K}) = \{\tilde{s}_1, \tilde{s}_2, \tilde{s}_{10}, \tilde{s}_{11}, \tilde{s}_{12}, \tilde{s}_{13}\}$, $DR_{WT}(\overline{K}) = \{\tilde{s}_3, \tilde{s}_6, \tilde{s}_7, \tilde{s}_8\}$ and $DR_V(\overline{K}) = \{\tilde{s}_4, \tilde{s}_5, \tilde{s}_9\}$.

The states are interpreted as: $\tilde{s}_1$ is the initial state in which the system is not activated; $\tilde{s}_2$: the system is activated and the memory is not requested and its maintenance is not initiated; $\tilde{s}_3$: the memory maintenance is initiated; $\tilde{s}_4$: the memory is requested by the first processor; $\tilde{s}_5$: the memory is requested by the second processor; $\tilde{s}_6$: the memory maintenance is initiated and the memory is requested by two processors; $\tilde{s}_7$: the memory maintenance is initiated and the memory is requested by the first processor; $\tilde{s}_8$: the memory maintenance is initiated and the memory is requested by the second processor; $\tilde{s}_9$: the memory is requested by two processors; $\tilde{s}_{10}$: the memory is allocated

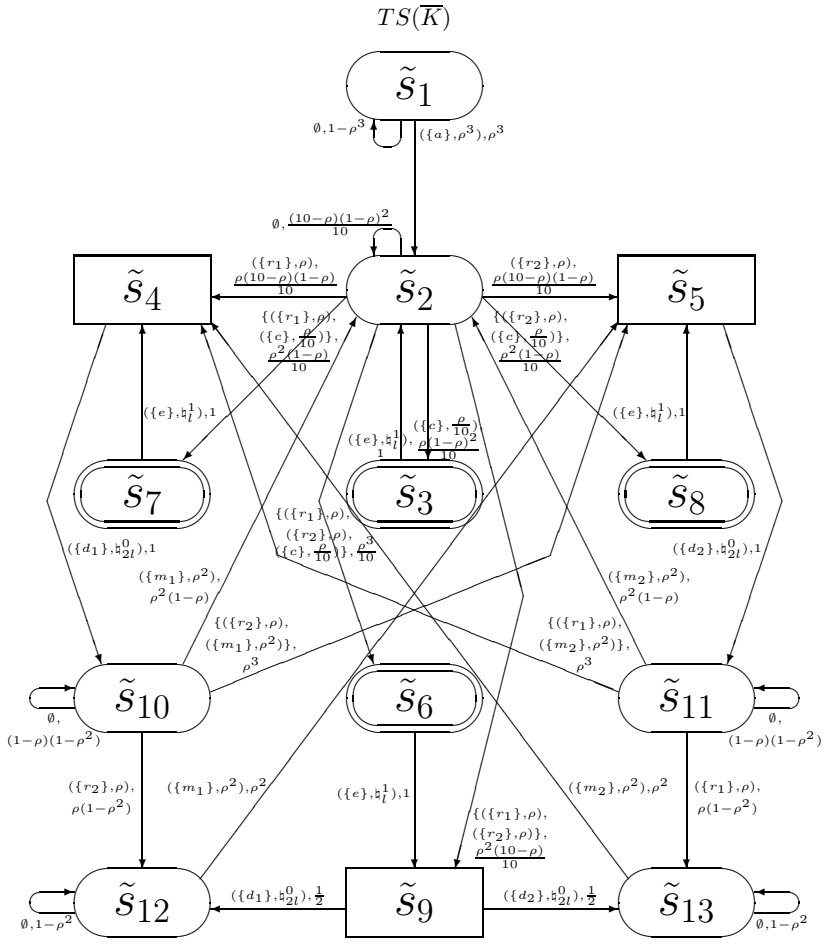


Figure 3. The transition system of the generalized shared memory system with maintenance.

to the first processor; $\tilde{s}_{11}$: the memory is allocated to the second processor; $\tilde{s}_{12}$: the memory is allocated to the first processor and requested by the second processor; $\tilde{s}_{13}$: the memory is allocated to the second processor and requested by the first processor.

In Figure 3, the transition system $TS(\overline{K})$ is presented. In Figure 4, the underlying SMC $SMC(\overline{K})$ is depicted. In step semantics, we can execute the following activities in parallel: $(\{r_1\},\rho),(\{r_2\},\rho)$, as well as $(\{r_1\},\rho),(\{m_2\},\rho^2)$, and $(\{r_2\},\rho),(\{m_1\},\rho^2)$. We can also execute in parallel $(\{r_1\},\rho),(\{c\},\frac{\rho}{10})$, as well as $(\{r_2\},\rho),(\{c\},\frac{\rho}{10})$, and even $(\{r_1\},\rho),(\{r_2\},\rho),(\{c\},\frac{\rho}{10})$. The states $\tilde{s}_6,\tilde{s}_7,\tilde{s}_8,\tilde{s}_9$ only exist in step semantics, since they are reachable exclusively by executing all three activities $(\{r_1\},\rho),(\{r_2\},\rho),(\{c\},\frac{\rho}{10})$ or any pair of them in parallel.

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

$$\widetilde{SJ} = \left(\tfrac{1}{\rho^3}, \tfrac{10}{\rho(21-12\rho+\rho^2)}, 1, 0, 0, 1, 1, 1, 0, \tfrac{1}{\rho(1+\rho-\rho^2)}, \tfrac{1}{\rho(1+\rho-\rho^2)}, \tfrac{1}{\rho^2}, \tfrac{1}{\rho^2}\right).$$

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

$$\widetilde{VAR} = \left(\tfrac{1-\rho^3}{\rho^6}, \tfrac{10(10-\rho)(1-\rho)^2}{\rho^2(21-12\rho+\rho^2)^2}, 0, 0, 0, 0, 0, 0, 0, \tfrac{(1-\rho^2)(1-\rho)}{\rho^2(1+\rho-\rho^2)^2}, \tfrac{(1-\rho^2)(1-\rho)}{\rho^2(1+\rho-\rho^2)^2}, \tfrac{1-\rho^2}{\rho^4}, \tfrac{1-\rho^2}{\rho^4}\right).$$

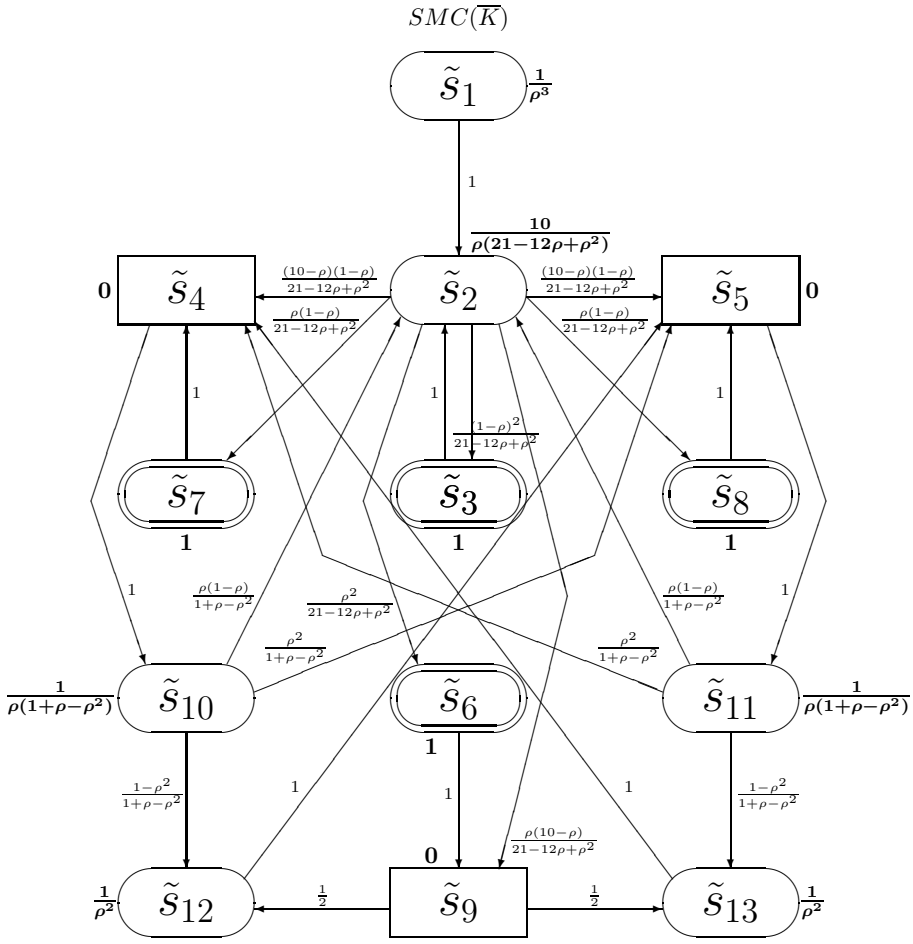


Figure 4. The underlying SMC of the generalized shared memory system with maintenance.

Let us denote $\chi = 21 - 12\rho + \rho^2$, $\theta = 1 + \rho - \rho^2$ and $\mu = 10 - \rho$. The TPM for $EDTMC(\overline{K})$ is

$$\widetilde{\mathbf{P}}^* = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{(1-\rho)^2}{\chi} & \frac{\mu(1-\rho)}{\chi} & \frac{\mu(1-\rho)}{\chi} & \frac{\rho^2}{\chi} & \frac{\rho(1-\rho)}{\chi} & \frac{\rho(1-\rho)}{\chi} & \frac{\rho\mu}{\chi} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{\rho(1-\rho)}{\theta} & 0 & 0 & \frac{\rho^2}{\theta} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-\rho^2}{\theta} & 0 \\ 0 & \frac{\rho(1-\rho)}{\theta} & 0 & \frac{\rho^2}{\theta} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-\rho^2}{\theta} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

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

$$\tilde{\psi}^* = \tfrac{1}{60+32\rho-94\rho^2+23\rho^3-\rho^4}(0, \rho(1-\rho)(21-12\rho+\rho^2), \rho(1-\rho)^3, 5(2-\rho)(1+\rho-\rho^2), \\ 5(2-\rho)(1+\rho-\rho^2), \rho^3(1-\rho), \rho^2(1-\rho)^2, \rho^2(1-\rho)^2, 10\rho^2(1-\rho), \\ 5(2-\rho)(1+\rho-\rho^2), 5(2-\rho)(1+\rho-\rho^2), 5(1-\rho)(2+\rho), 5(1-\rho)(2+\rho)).$$

The steady-state PMF $\tilde{\psi}^*$ weighted by $\widetilde{SJ}$ is

$$\tfrac{1}{\rho^2(60+32\rho-94\rho^2+23\rho^3-\rho^4)}(0, 10\rho^2(1-\rho), \rho^3(1-\rho)^3, 0, 0, \rho^5(1-\rho), \rho^4(1-\rho)^2, \\ \rho^4(1-\rho)^2, 0, 5\rho(2-\rho), 5\rho(2-\rho), 5(1-\rho)(2+\rho), 5(1-\rho)(2+\rho)).$$

We normalize the steady-state weighted PMF by dividing it by its components sum

$$\tilde{\psi}^* \widetilde{SJ}^T = \frac{20 + 10\rho - 10\rho^2 - 9\rho^3 - \rho^4}{\rho^2(60 + 32\rho - 94\rho^2 + 23\rho^3 - \rho^4)}.$$

Thus, the steady-state PMF for $SMC(\overline{K})$ is

$$\tilde{\varphi} = \tfrac{1}{20+10\rho-10\rho^2-9\rho^3-\rho^4}(0, 10\rho^2(1-\rho), \rho^3(1-\rho)^3, 0, 0, \rho^5(1-\rho), \rho^4(1-\rho)^2, \\ \rho^4(1-\rho)^2, 0, 5\rho(2-\rho), 5\rho(2-\rho), 5(1-\rho)(2+\rho), 5(1-\rho)(2+\rho)).$$

Otherwise, from $TS(\overline{K})$, we can construct the DTMC of $\overline{K}$, $DTMC(\overline{K})$, and then calculate $\tilde{\varphi}$ using it. In Figure 5, the DTMC $DTMC(\overline{K})$ is depicted.

Let us denote $\mu = 10 - \rho$. The TPM for $DTMC(\overline{K})$ is

$$\widetilde{\mathbf{P}} = \begin{pmatrix} 1-\rho^3 & \rho^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\mu(1-\rho)^2}{10} & \frac{\rho(1-\rho)^2}{10} & \frac{\rho\mu(1-\rho)}{10} & \frac{\rho\mu(1-\rho)}{10} & \frac{\rho^3}{10} & \frac{\rho^2(1-\rho)}{10} & \frac{\rho^2(1-\rho)}{10} & \frac{\rho^2\mu}{10} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \rho^2(1-\rho) & 0 & 0 & \rho^3 & 0 & 0 & 0 & 0 & (1-\rho)(1-\rho^2) & 0 & \rho(1-\rho^2) & 0 \\ 0 & \rho^2(1-\rho) & 0 & \rho^3 & 0 & 0 & 0 & 0 & 0 & 0 & (1-\rho)(1-\rho^2) & 0 & \rho(1-\rho^2) \\ 0 & 0 & 0 & 0 & \rho^2 & 0 & 0 & 0 & 0 & 0 & 0 & 1-\rho^2 & 0 \\ 0 & 0 & 0 & \rho^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-\rho^2 \end{pmatrix}.$$

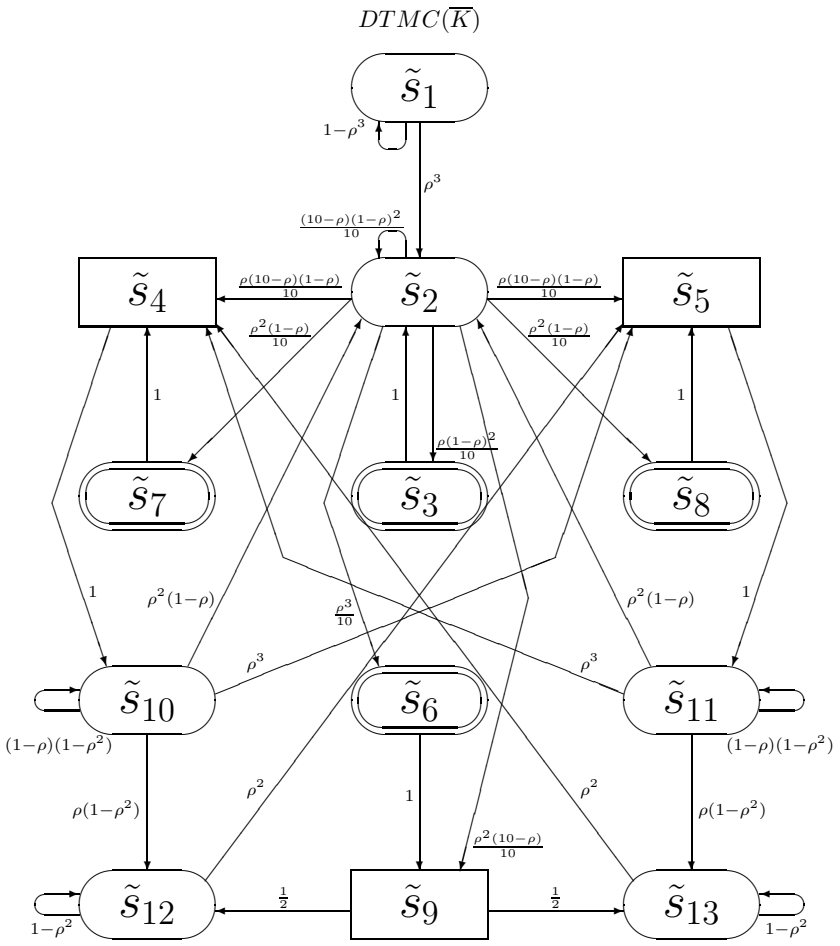


Figure 5. The DTMC of the generalized shared memory system with maintenance.

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

$$\tilde{\psi} = \tfrac{1}{20+10\rho+10\rho^2+\rho^3-21\rho^4}(0, 10\rho^2(1-\rho), \rho^3(1-\rho)^3, 5\rho^2(2-\rho)(1+\rho-\rho^2),$$
$$5\rho^2(2-\rho)(1+\rho-\rho^2), \rho^5(1-\rho), \rho^4(1-\rho)^2, \rho^4(1-\rho)^2, 10\rho^4(1-\rho), 5\rho(2-\rho),$$
$$5\rho(2-\rho), 5(1-\rho)(2+\rho), 5(1-\rho)(2+\rho)).$$

Remember that $DR_T(\overline{K}) = DR_{ST}(\overline{K}) \cup DR_{WT}(\overline{K}) = \{\tilde{s}_1, \tilde{s}_2, \tilde{s}_3, \tilde{s}_6, \tilde{s}_7, \tilde{s}_8, \tilde{s}_{10}, \tilde{s}_{11}, \tilde{s}_{12}, \tilde{s}_{13}\}$ and $DR_V(\overline{K}) = \{\tilde{s}_4, \tilde{s}_5, \tilde{s}_9\}$. Hence,

$$\textstyle\sum_{\tilde{s}\in DR_T(\overline{K})} \tilde{\psi}(\tilde{s}) = \tilde{\psi}(\tilde{s}_1) + \tilde{\psi}(\tilde{s}_2) + \tilde{\psi}(\tilde{s}_3) + \tilde{\psi}(\tilde{s}_6) + \tilde{\psi}(\tilde{s}_7) + \tilde{\psi}(\tilde{s}_8) + \tilde{\psi}(\tilde{s}_{10}) +$$
$$\tilde{\psi}(\tilde{s}_{11}) + \tilde{\psi}(\tilde{s}_{12}) + \tilde{\psi}(\tilde{s}_{13}) = \tfrac{20+10\rho-10\rho^2-9\rho^3-\rho^4}{20+10\rho+10\rho^2+\rho^3-21\rho^4}.$$

By Proposition 4 from [40] (the 'non-quotient' analogue of Proposition 5.6), we have

$$\tilde{\varphi}(\tilde{s}_1) = 0 \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = 0,$$
$$\tilde{\varphi}(\tilde{s}_2) = \tfrac{10\rho^2(1-\rho)}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4},$$
$$\tilde{\varphi}(\tilde{s}_3) = \tfrac{\rho^3(1-\rho)^3}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{\rho^3(1-\rho)^3}{20+10\rho-10\rho^2-9\rho^3-\rho^4},$$
$$\tilde{\varphi}(\tilde{s}_4) = 0,$$
$$\tilde{\varphi}(\tilde{s}_5) = 0,$$
$$\tilde{\varphi}(\tilde{s}_6) = \tfrac{\rho^5(1-\rho)}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{\rho^5(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4},$$
$$\tilde{\varphi}(\tilde{s}_7) = \tfrac{\rho^4(1-\rho)^2}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{\rho^4(1-\rho)^2}{20+10\rho-10\rho^2-9\rho^3-\rho^4},$$
$$\tilde{\varphi}(\tilde{s}_8) = \tfrac{\rho^4(1-\rho)^2}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{\rho^4(1-\rho)^2}{20+10\rho-10\rho^2-9\rho^3-\rho^4},$$
$$\tilde{\varphi}(\tilde{s}_9) = 0,$$
$$\tilde{\varphi}(\tilde{s}_{10}) = \tfrac{5\rho(2-\rho)}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{5\rho(2-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4},$$
$$\tilde{\varphi}(\tilde{s}_{11}) = \tfrac{5\rho(2-\rho)}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{5\rho(2-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4},$$
$$\tilde{\varphi}(\tilde{s}_{12}) = \tfrac{5(1-\rho)(2+\rho)}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{5(1-\rho)(2+\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4},$$
$$\tilde{\varphi}(\tilde{s}_{13}) = \tfrac{5(1-\rho)(2+\rho)}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{5(1-\rho)(2+\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}.$$

Thus, the steady-state PMF for $SMC(\overline{K})$ is

$$\tilde{\varphi} = \tfrac{1}{20+10\rho-10\rho^2-9\rho^3-\rho^4}(0, 10\rho^2(1-\rho), \rho^3(1-\rho)^3, 0, 0, \rho^5(1-\rho), \rho^4(1-\rho)^2,$$
$$\rho^4(1-\rho)^2, 0, 5\rho(2-\rho), 5\rho(2-\rho), 5(1-\rho)(2+\rho), 5(1-\rho)(2+\rho)).$$

This coincides with the result obtained with the use of $\tilde{\psi}^*$ and $\widetilde{SJ}$.

Alternatively, from $TS(\overline{K})$, we can construct the reduced DTMC of $\overline{K}$, $RDTMC(\overline{K})$, and then calculate $\tilde{\varphi}$ using it.

Remember that $DR_{ST}(\overline{K}) = \{\tilde{s}_1, \tilde{s}_2, \tilde{s}_{10}, \tilde{s}_{11}, \tilde{s}_{12}, \tilde{s}_{13}\}$, $DR_{WT}(\overline{K}) = \{\tilde{s}_3, \tilde{s}_6, \tilde{s}_7, \tilde{s}_8\}$, $DR_V(\overline{K}) = \{\tilde{s}_4, \tilde{s}_5, \tilde{s}_9\}$. We reorder the elements of $DR(\overline{K})$, by moving vanishing states to the first positions and s-tangible states to the last positions: $\tilde{s}_4, \tilde{s}_5, \tilde{s}_9, \tilde{s}_3, \tilde{s}_6, \tilde{s}_7, \tilde{s}_8, \tilde{s}_1, \tilde{s}_2, \tilde{s}_{10}, \tilde{s}_{11}, \tilde{s}_{12}, \tilde{s}_{13}$.

Let us denote $\mu = 10 - \rho$. The reordered TPM for $DTMC(\overline{K})$ is

$$\widetilde{\mathbf{P}}_r = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-\rho^3 & \rho^3 & 0 & 0 & 0 & 0 \\
\frac{\rho\mu(1-\rho)}{10} & \frac{\rho\mu(1-\rho)}{10} & \frac{\rho^2\mu}{10} & \frac{\rho(1-\rho)^2}{10} & \frac{\rho^3}{10} & \frac{\rho^2(1-\rho)}{10} & \frac{\rho^2(1-\rho)}{10} & 0 & \frac{\mu(1-\rho)^2}{10} & 0 & 0 & 0 & 0 \\
0 & \rho^3 & 0 & 0 & 0 & 0 & 0 & 0 & \rho^2(1-\rho) & (1-\rho)(1-\rho^2) & 0 & \rho(1-\rho^2) & 0 \\
\rho^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \rho^2(1-\rho) & 0 & (1-\rho)(1-\rho^2) & 0 & \rho(1-\rho^2) \\
0 & \rho^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-\rho^2 & 0 \\
\rho^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-\rho^2
\end{pmatrix}.$$

The result of the decomposing $\widetilde{\mathbf{P}}_r$ are the matrices

$$\widetilde{\mathbf{C}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \ \widetilde{\mathbf{D}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \ \widetilde{\mathbf{E}} = \begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0 \\
\frac{\rho\mu(1-\rho)}{10} & \frac{\rho\mu(1-\rho)}{10} & \frac{\rho^2\mu}{10} \\
0 & \rho^3 & 0 \\
\rho^3 & 0 & 0 \\
0 & \rho^2 & 0 \\
\rho^2 & 0 & 0
\end{pmatrix},$$

$$\widetilde{\mathbf{F}} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1-\rho^3 & \rho^3 & 0 & 0 & 0 & 0 \\
\frac{\rho(1-\rho)^2}{10} & \frac{\rho^3}{10} & \frac{\rho^2(1-\rho)}{10} & \frac{\rho^2(1-\rho)}{10} & 0 & \frac{\mu(1-\rho)^2}{10} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \rho^2(1-\rho) & (1-\rho)(1-\rho^2) & 0 & \rho(1-\rho^2) & 0 \\
0 & 0 & 0 & 0 & 0 & \rho^2(1-\rho) & 0 & (1-\rho)(1-\rho^2) & 0 & \rho(1-\rho^2) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-\rho^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-\rho^2
\end{pmatrix}.$$

Since $\widetilde{\mathbf{C}}^1 = \mathbf{0}$, we have $\forall k > 0, \ \widetilde{\mathbf{C}}^k = \mathbf{0}$, hence, $l = 0$ and there are no loops among vanishing states. Then

$$\widetilde{\mathbf{G}} = \sum_{k=0}^{l} \widetilde{\mathbf{C}}^k = \widetilde{\mathbf{C}}^0 = \mathbf{I}.$$

Further, the TPM for $RDTMC(\overline{K})$ is

$$\widetilde{\mathbf{P}}^{\diamond} = \widetilde{\mathbf{F}} + \widetilde{\mathbf{E}}\widetilde{\mathbf{G}}\widetilde{\mathbf{D}} = \widetilde{\mathbf{F}} + \widetilde{\mathbf{E}}\mathbf{I}\widetilde{\mathbf{D}} = \widetilde{\mathbf{F}} + \widetilde{\mathbf{E}}\widetilde{\mathbf{D}} =$$

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1-\rho^3 & \rho^3 & 0 & 0 & 0 & 0 \\
\frac{\rho(1-\rho)^2}{10} & \frac{\rho^3}{10} & \frac{\rho^2(1-\rho)}{10} & \frac{\rho^2(1-\rho)}{10} & 0 & \frac{\mu(1-\rho)^2}{10} & \frac{\rho\mu(1-\rho)}{10} & \frac{\rho\mu(1-\rho)}{10} & \frac{\rho^2\mu}{20} & \frac{\rho^2\mu}{20} \\
0 & 0 & 0 & 0 & 0 & \rho^2(1-\rho) & (1-\rho)(1-\rho^2) & \rho^3 & \rho(1-\rho^2) & 0 \\
0 & 0 & 0 & 0 & 0 & \rho^2(1-\rho) & \rho^3 & (1-\rho)(1-\rho^2) & 0 & \rho(1-\rho^2) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \rho^2 & 1-\rho^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \rho^2 & 0 & 0 & 1-\rho^2
\end{pmatrix}.$$

In Figure 6, the reduced DTMC $RDTMC(\overline{K})$ is presented.
Then the steady-state PMF for $RDTMC(\overline{K})$ is

$$\tilde{\psi}^{\diamond} = \tfrac{1}{20+10\rho-10\rho^2-9\rho^3-\rho^4}(\rho^3(1-\rho)^3, \rho^5(1-\rho), \rho^4(1-\rho)^2, \rho^4(1-\rho)^2, 0, \\ 10\rho^2(1-\rho), 5\rho(2-\rho), 5\rho(2-\rho), 5(1-\rho)(2+\rho), 5(1-\rho)(2+\rho)).$$

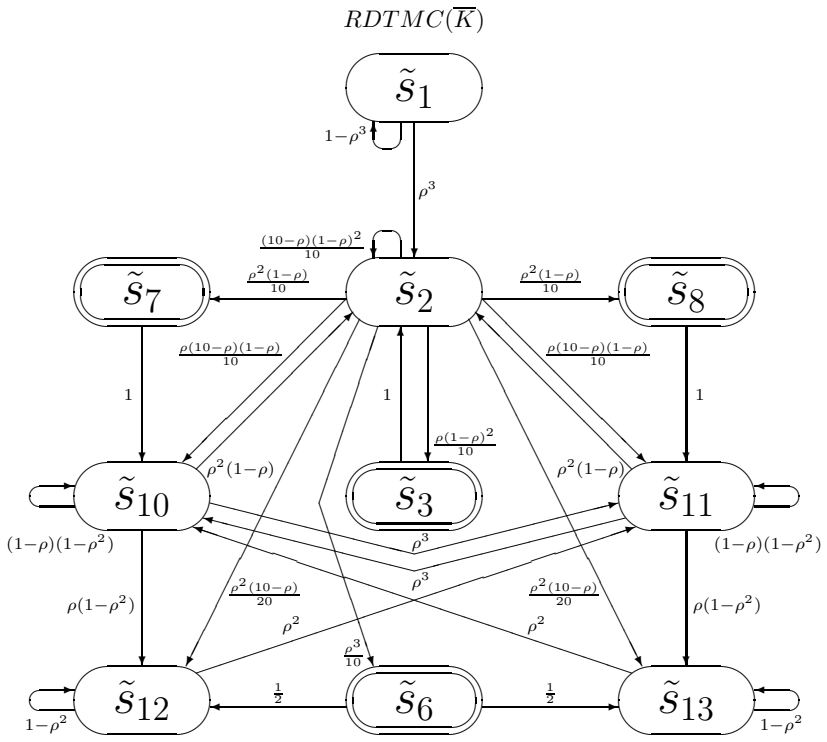


Figure 6. The reduced DTMC of the generalized shared memory system with maintenance.

Note that $\tilde{\psi}^{\diamond} = (\tilde{\psi}^{\diamond}(\tilde{s}_3), \tilde{\psi}^{\diamond}(\tilde{s}_6), \tilde{\psi}^{\diamond}(\tilde{s}_7), \tilde{\psi}^{\diamond}(\tilde{s}_8), \tilde{\psi}^{\diamond}(\tilde{s}_1), \tilde{\psi}^{\diamond}(\tilde{s}_2), \tilde{\psi}^{\diamond}(\tilde{s}_{10}), \tilde{\psi}^{\diamond}(\tilde{s}_{11}), \tilde{\psi}^{\diamond}(\tilde{s}_{12}), \tilde{\psi}^{\diamond}(\tilde{s}_{13}))$.

By Proposition 5 from [40] (the 'non-quotient' analogue of Proposition 5.8), we have

$$\begin{aligned}
&\tilde{\varphi}(\tilde{s}_1) = 0, && \tilde{\varphi}(\tilde{s}_2) = \tfrac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4},\\
&\tilde{\varphi}(\tilde{s}_3) = \tfrac{\rho^3(1-\rho)^3}{20+10\rho-10\rho^2-9\rho^3-\rho^4}, && \tilde{\varphi}(\tilde{s}_4) = 0,\\
&\tilde{\varphi}(\tilde{s}_5) = 0, && \tilde{\varphi}(\tilde{s}_6) = \tfrac{\rho^5(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4},\\
&\tilde{\varphi}(\tilde{s}_7) = \tfrac{\rho^4(1-\rho)^2}{20+10\rho-10\rho^2-9\rho^3-\rho^4}, && \tilde{\varphi}(\tilde{s}_8) = \tfrac{\rho^4(1-\rho)^2}{20+10\rho-10\rho^2-9\rho^3-\rho^4},\\
&\tilde{\varphi}(\tilde{s}_9) = 0, && \tilde{\varphi}(\tilde{s}_{10}) = \tfrac{5\rho(2-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4},\\
&\tilde{\varphi}(\tilde{s}_{11}) = \tfrac{5\rho(2-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}, && \tilde{\varphi}(\tilde{s}_{12}) = \tfrac{(1-\rho)(2+\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4},\\
&\tilde{\varphi}(\tilde{s}_{13}) = \tfrac{(1-\rho)(2+\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}.
\end{aligned}$$

Thus, the steady-state PMF for $SMC(\overline{K})$ is

$$\tilde{\varphi} = \tfrac{1}{20+10\rho-10\rho^2-9\rho^3-\rho^4}(0, 10\rho^2(1-\rho), \rho^3(1-\rho)^3, 0, 0, \rho^5(1-\rho), \rho^4(1-\rho)^2,\\ \rho^4(1-\rho)^2, 0, 5\rho(2-\rho), 5\rho(2-\rho), 5(1-\rho)(2+\rho), 5(1-\rho)(2+\rho)).$$

This coincides with the result obtained with the use of $\tilde{\psi}^*$ and $\widetilde{SJ}$.
We can now calculate the main performance indices.

- The average recurrence time in the state $\tilde{s}_2$, where no processor requests the memory and its maintenance is not initiated, called the *average system run-through*, is $\frac{1}{\tilde{\varphi}_2} = \frac{20+10\rho-10\rho^2-9\rho^3-\rho^4}{10\rho^2(1-\rho)}$.

- The system is not activated only in the state $\tilde{s}_1$. Then the steady-state probability that the system is activated is $1-\tilde{\varphi}_1 = 1-0 = 1$. The common memory is only available in the states $\tilde{s}_2, \tilde{s}_4, \tilde{s}_5, \tilde{s}_9$. Then the steady-state probability that the memory is available is $\tilde{\varphi}_2+\tilde{\varphi}_4+\tilde{\varphi}_5+\tilde{\varphi}_9 = \frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}+0+0+0 = \frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$. The common memory is maintained only in the states $\tilde{s}_3, \tilde{s}_6, \tilde{s}_7, \tilde{s}_8$. Then the steady-state probability that the memory is maintained is $\tilde{\varphi}_3+\tilde{\varphi}_6+\tilde{\varphi}_7+\tilde{\varphi}_8 = \frac{\rho^3(1-\rho)^3}{20+10\rho-10\rho^2-9\rho^3-\rho^4}+\frac{\rho^5(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}+\frac{\rho^4(1-\rho)^2}{20+10\rho-10\rho^2-9\rho^3-\rho^4}+\frac{\rho^4(1-\rho)^2}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \frac{\rho^3(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$. The steady-state probability that the memory is used (i.e. neither available nor maintained), called the *shared memory utilization*, is $1-\frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}-\frac{\rho^3(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \frac{10(2+\rho-2\rho^2)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$.
- After activation of the system, we leave the state $\tilde{s}_1$ for ever, and the common memory is either requested or allocated or maintained in every remaining state, with exception of $\tilde{s}_2$. Thus, the *rate with which the necessity (also for maintenance) of shared memory emerges* coincides with the rate of leaving $\tilde{s}_2$, calculated as $\frac{\tilde{\varphi}_2}{\widetilde{SJ}_2} = \frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}\cdot\frac{\rho(21-12\rho+\rho^2)}{10} = \frac{\rho^3(1-\rho)(21-12\rho+\rho^2)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$.
- The parallel common memory request of two processors $\{(\{r_1\},\rho),(\{r_2\},\rho)\}$ is only possible from the state $\tilde{s}_2$. In this state, the request probability is the sum of the execution probabilities for all multisets of activities containing both $(\{r_1\},\rho)$ and $(\{r_2\},\rho)$. The *steady-state probability of the shared memory request from two processors* is $\tilde{\varphi}_2\sum_{\{\Upsilon|(\{(\{r_1\},\rho),(\{r_2\},\rho)\}\subseteq\Upsilon\}}PT(\Upsilon,\tilde{s}_2) = \frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}\left(\frac{\rho^2(10-\rho)}{10}+\frac{\rho^3}{10}\right) = \frac{10\rho^4(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$.
- The common memory request of the first processor $(\{r_1\},\rho)$ is possible from the states $\tilde{s}_2, \tilde{s}_{11}$. In each of them, the request probability is the sum of the execution probabilities for all sets of activities containing $(\{r_1\},\rho)$. The *steady-state probability of the shared memory request from the first processor* is $\tilde{\varphi}_2\sum_{\{\Upsilon|(\{r_1\},\rho)\in\Upsilon\}}PT(\Upsilon,\tilde{s}_2)+\tilde{\varphi}_{11}\sum_{\{\Upsilon|(\{r_1\},\rho)\in\Upsilon\}}PT(\Upsilon,\tilde{s}_{11}) = \frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}\left(\frac{\rho(10-\rho)(1-\rho)}{10}+\frac{\rho^2(1-\rho)}{10}+\frac{\rho^2(10-\rho)}{10}+\frac{\rho^3}{10}\right)+\frac{5\rho(2-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}(\rho(1-\rho^2)+\rho^3) = \frac{5\rho^2(2+\rho-2\rho^2)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$.

In Figure 7, the marked dtsd-boxes corresponding to the dynamic expressions of the generalized two processors, shared memory and shared memory system with maintenance are presented, i.e. $N_i = Box_{dtsd}(\overline{K_i})$ $(1\le i\le 3)$ and $N = Box_{dtsd}(\overline{K})$.

7.2. The abstract system and its reduction

Consider a modification of the generalized shared memory system with maintenance via abstraction from identifiers of the processors, i.e. such that the processors are indistinguishable. We can see that a processor requires memory or the memory is allocated to it but cannot observe which processor is it. We call this system the abstract generalized shared memory system with maintenance. To implement the abstraction, we replace the actions r_i, d_i, m_i $(1\le i\le 2)$ in the system specification by r, d, m, respectively.

The static expression of the first processor is

$$L_1 = [(\{x_1\},\rho)*((\{r\},\rho);(\{d,y_1\},\natural_l^0);(\{m,z_1\},\rho))*\mathsf{Stop}].$$

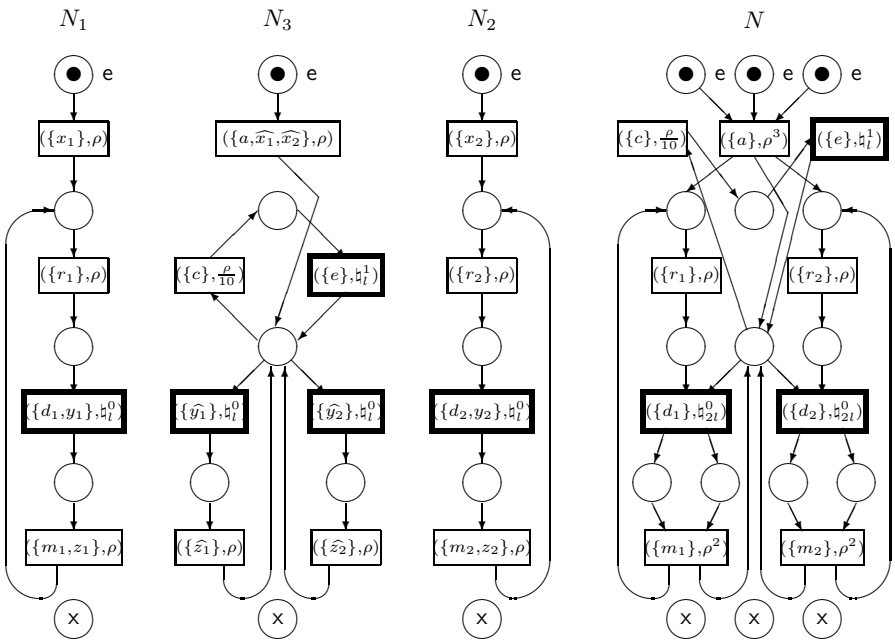


Figure 7. The marked dtsd-boxes of the generalized two processors, shared memory and shared memory system with maintenance.

The static expression of the second processor is

$$L_2 = [(\{x_2\},\rho) * ((\{r\},\rho);(\{d,y_2\},\natural_l^0);(\{m,z_2\},\rho)) * \mathsf{Stop}].$$

The static expression of the shared memory is

$$L_3 = [(\{a,\widehat{x_1},\widehat{x_2}\},\rho) * ((((\{c\},\tfrac{\rho}{10});(\{e\},\natural_l^1))[]((\{\widehat{y_1}\},\natural_l^0);(\{\widehat{z_1}\},\rho))[] \\ ((\{\widehat{y_2}\},\natural_l^0);(\{\widehat{z_2}\},\rho))) * \mathsf{Stop}].$$

The static expression of the abstract generalized shared memory system with maintenance is

$$L = (L_1 \| L_2 \| L_3)\ \mathsf{sr}\ (x_1,x_2,y_1,y_2,z_1,z_2).$$

$DR(\overline{L}) = \{\tilde{s}'_1,\ldots,\tilde{s}'_{13}\}$ resembles $DR(\overline{K})$, and $TS(\overline{L})$ is similar to $TS(\overline{K})$. Since $SMC(\overline{L}) \simeq SMC(\overline{K})$, the average sojourn time vectors of $\overline{L}$ and $\overline{K}$, the TPMs and the steady-state PMFs for $EDTMC(\overline{L})$ and $EDTMC(\overline{K})$ coincide.

The first, second, third and fourth performance indices are the same for the generalized system and its abstract modification. Let us consider the following performance index which is specific to the abstract system.

- The common memory request of a processor $(\{r\},\rho)$ is possible from the states $\tilde{s}'_2,\tilde{s}'_{10},\tilde{s}'_{11}$. In each of them, the request probability is the sum of the execution probabilities for all sets of activities containing $(\{r\},\rho)$. The *steady-state probability of the shared memory request from a processor* is $\tilde{\varphi}_2 \sum_{\{\Upsilon|(\{r\},\rho)\in\Upsilon\}} PT(\Upsilon,\tilde{s}'_2) + \tilde{\varphi}_{10} \sum_{\{\Upsilon|(\{r\},\rho)\in\Upsilon\}} PT(\Upsilon,\tilde{s}'_{10}) +$
$\tilde{\varphi}_{11} \sum_{\{\Upsilon|(\{r\},\rho)\in\Upsilon\}} PT(\Upsilon,\tilde{s}'_{11}) = \frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4} \left(\frac{\rho(10-\rho)(1-\rho)}{10} + \frac{\rho(10-\rho)(1-\rho)}{10} + \right.$

$\frac{\rho^2(1-\rho)}{10} + \frac{\rho^2(1-\rho)}{10} + \frac{\rho^2(10-\rho)}{10} + \frac{\rho^3}{10}\Big) + \frac{5\rho(2-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}(\rho(1-\rho^2)+\rho^3) + \frac{5\rho(2-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}(\rho(1-\rho^2)+\rho^3) = \frac{10\rho^2(2-\rho)(1+\rho-\rho^2)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$.

We have $DR(\overline{L})/_{\mathcal{R}_{ss}(\overline{L})} = \{\widetilde{\mathcal{K}}_1, \widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}_3, \widetilde{\mathcal{K}}_4, \widetilde{\mathcal{K}}_5, \widetilde{\mathcal{K}}_6, \widetilde{\mathcal{K}}_7, \widetilde{\mathcal{K}}_8, \widetilde{\mathcal{K}}_9\}$, where $\widetilde{\mathcal{K}}_1 = \{\tilde{s}'_1\}$ (the initial state in which the system is not activated), $\widetilde{\mathcal{K}}_2 = \{\tilde{s}'_2\}$ (the system is activated and the memory is not requested and its maintenance is not initiated), $\widetilde{\mathcal{K}}_3 = \{\tilde{s}'_3\}$ (the memory maintenance is initiated), $\widetilde{\mathcal{K}}_4 = \{\tilde{s}'_4, \tilde{s}'_5\}$ (the memory is requested by a processor), $\widetilde{\mathcal{K}}_5 = \{\tilde{s}'_6\}$ (the memory maintenance is initiated and the memory is requested by two processors), $\widetilde{\mathcal{K}}_6 = \{\tilde{s}'_7, \tilde{s}'_8\}$ (the memory maintenance is initiated and the memory is requested by a processor), $\widetilde{\mathcal{K}}_7 = \{\tilde{s}'_9\}$ (the memory is requested by two processors), $\widetilde{\mathcal{K}}_8 = \{\tilde{s}'_{10}, \tilde{s}'_{11}\}$ (the memory is allocated to a processor), $\widetilde{\mathcal{K}}_9 = \{\tilde{s}'_{12}, \tilde{s}'_{13}\}$ (the memory is allocated to a processor and requested by another processor).

We have $DR_{ST}(\overline{L})/_{\mathcal{R}_{ss}(\overline{L})} = \{\widetilde{\mathcal{K}}_1, \widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}_8, \widetilde{\mathcal{K}}_9\}$, $DR_{WT}(\overline{L})/_{\mathcal{R}_{ss}(\overline{L})} = \{\widetilde{\mathcal{K}}_3, \widetilde{\mathcal{K}}_5, \widetilde{\mathcal{K}}_6\}$, $DR_V(\overline{L})/_{\mathcal{R}_{ss}(\overline{L})} = \{\widetilde{\mathcal{K}}_4, \widetilde{\mathcal{K}}_7\}$.

In Figure 8, the quotient transition system $TS_{\underline{\leftrightarrow}_{ss}}(\overline{L})$ is presented. In Figure 9, the quotient underlying SMC $SMC_{\underline{\leftrightarrow}_{ss}}(\overline{L})$ is depicted. Note that, in step semantics, we may execute the following multiactions in parallel: $\{r\}, \{r\}$, as well as $\{r\}, \{m\}$. We can also execute in parallel $\{r\}, \{c\}$, and even $\{r\}, \{r\}, \{c\}$. The states $\widetilde{\mathcal{K}}_5, \widetilde{\mathcal{K}}_6, \widetilde{\mathcal{K}}_7$ only exist in step semantics, since they are reachable exclusively by executing all three multiactions $\{r\}, \{r\}, \{c\}$ or any pair of them in parallel.

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

$$\widetilde{SJ}' = \left(\frac{1}{\rho^3}, \frac{10}{\rho(21-12\rho+\rho^2)}, 1, 0, 1, 1, 0, \frac{1}{\rho(1+\rho-\rho^2)}, \frac{1}{\rho^2}\right).$$

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

$$\widetilde{VAR}' = \left(\frac{1-\rho^3}{\rho^6}, \frac{10(10-\rho)(1-\rho)^2}{\rho^2(21-12\rho+\rho^2)^2}, 0, 0, 0, 0, 0, \frac{(1-\rho^2)(1-\rho)}{\rho^2(1+\rho-\rho^2)^2}, \frac{1-\rho^2}{\rho^4}\right).$$

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

$$\widetilde{\mathbf{P}}'^{*} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{(1-\rho)^2}{21-12\rho+\rho^2} & \frac{2(10-\rho)(1-\rho)}{21-12\rho+\rho^2} & \frac{\rho^2}{21-12\rho+\rho^2} & \frac{2\rho(1-\rho)}{21-12\rho+\rho^2} & \frac{\rho(10-\rho)}{21-12\rho+\rho^2} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{\rho(1-\rho)}{1+\rho-\rho^2} & 0 & \frac{\rho^2}{1+\rho-\rho^2} & 0 & 0 & 0 & 0 & \frac{1-\rho^2}{1+\rho-\rho^2} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

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

$$\tilde{\psi}'^{*} = \frac{1}{60+32\rho-94\rho^2+23\rho^3-\rho^4}(0, \rho(1-\rho)(21-12\rho+\rho^2), \rho(1-\rho)^3, 10(2-\rho)(1+\rho-\rho^2), \rho^3(1-\rho), 2\rho^2(1-\rho)^2, 10\rho^2(1-\rho), 10(2-\rho)(1+\rho-\rho^2), 10(1-\rho)(2+\rho)).$$

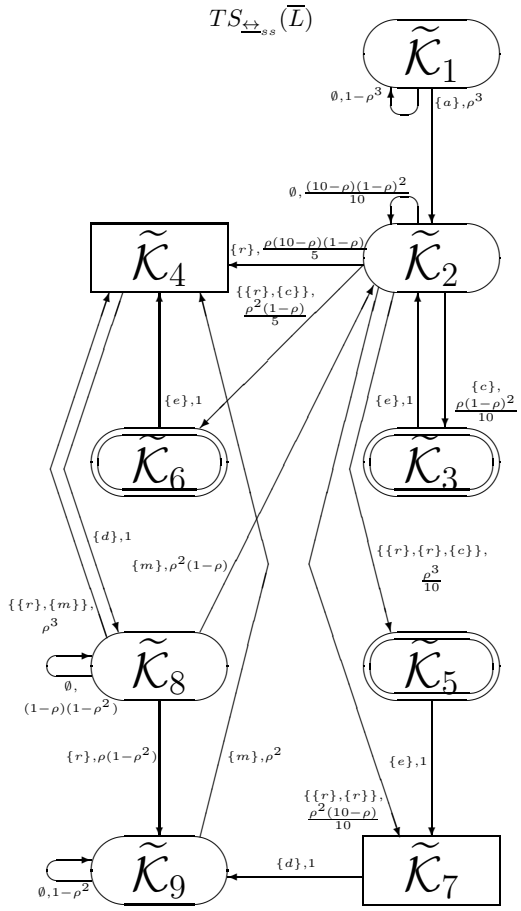


Figure 8. The quotient transition system of the abstract generalized shared memory system with maintenance.

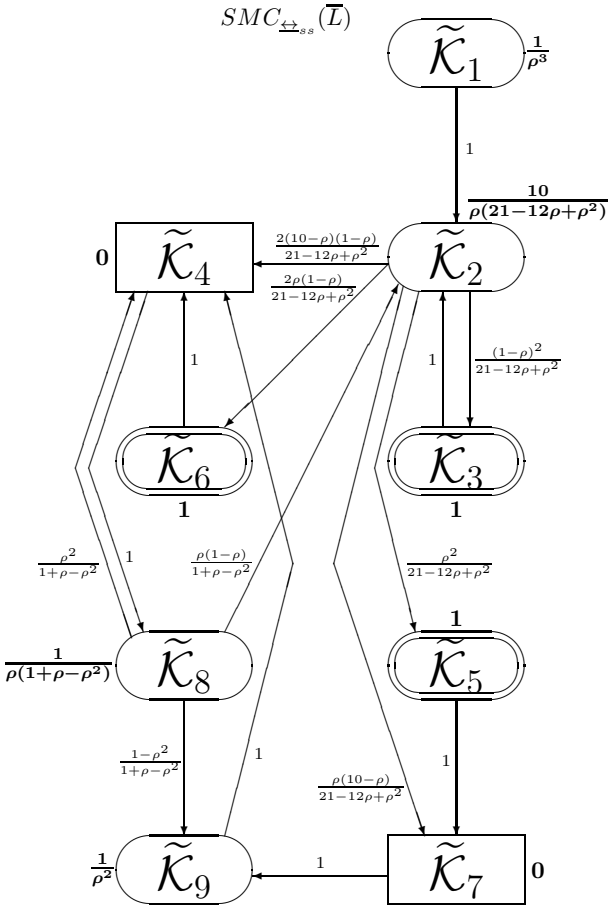


Figure 9. The quotient underlying SMC of the abstract generalized shared memory system with maintenance.

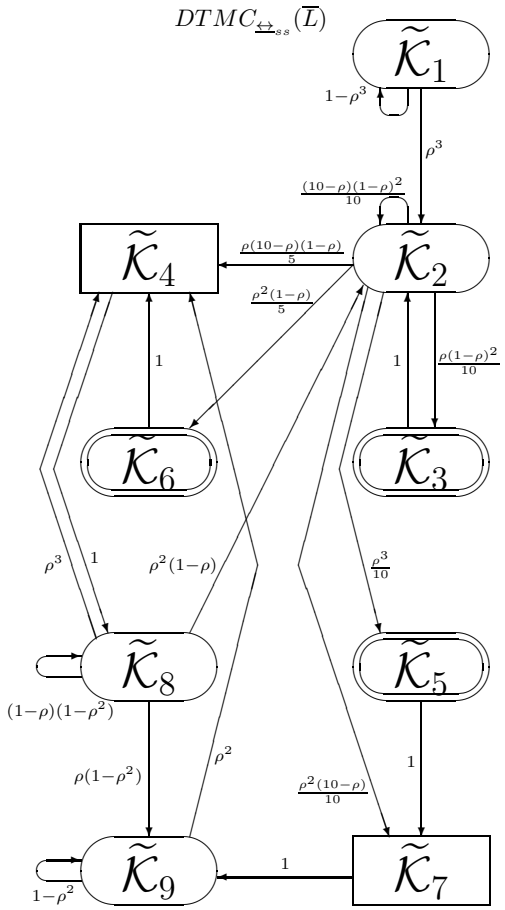


Figure 10. The quotient DTMC of the abstract generalized shared memory system with maintenance.

The steady-state PMF $\tilde{\psi}'^*$ weighted by $\widetilde{SJ}'$ is

$$\frac{1}{60+32\rho-94\rho^2+23\rho^3-\rho^4}(0, 10(1-\rho), \rho(1-\rho)^3, 0, \rho^3(1-\rho), 2\rho^2(1-\rho)^2, 0, 10(2-\rho), \\ 10(1-\rho)(2+\rho)).$$

We normalize the steady-state weighted PMF by dividing it by its components sum

$$\tilde{\psi}'^* \widetilde{SJ}'^T = \frac{20+10\rho-10\rho^2-9\rho^3-\rho^4}{\rho^2(60+32\rho-94\rho^2+23\rho^3-\rho^4)}.$$

Thus, the steady-state PMF for $SMC_{\underline{\leftrightarrow}_{ss}}(\overline{L})$ is

$$\tilde{\varphi}' = \frac{1}{20+10\rho-10\rho^2-9\rho^3-\rho^4}(0, 10\rho^2(1-\rho), \rho^3(1-\rho)^3, 0, \rho^5(1-\rho), 2\rho^4(1-\rho)^2, 0, \\ 10\rho(2-\rho), 10(1-\rho)(2+\rho)).$$

Otherwise, from $TS_{\underline{\leftrightarrow}_{ss}}(\overline{L})$, we can build the quotient DTMC of $\overline{L}$, $DTMC_{\underline{\leftrightarrow}_{ss}}(\overline{L})$, and calculate $\tilde{\varphi}'$ using it. In Figure 10, the quotient DTMC $DTMC_{\underline{\leftrightarrow}_{ss}}(\overline{L})$ is depicted.

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

$$\widetilde{\mathbf{P}}'=\begin{pmatrix} 1-\rho^3 & \rho^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{(10-\rho)(1-\rho)^2}{10} & \frac{\rho(1-\rho)^2}{10} & \frac{\rho(10-\rho)(1-\rho)}{5} & \frac{\rho^3}{10} & \frac{\rho^2(1-\rho)}{5} & \frac{\rho^2(10-\rho)}{10} & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \rho^2(1-\rho) & 0 & \rho^3 & 0 & 0 & 0 & (1-\rho)(1-\rho^2) & \rho(1-\rho^2) \\ 0 & 0 & 0 & \rho^2 & 0 & 0 & 0 & 0 & 1-\rho^2 \end{pmatrix}.$$

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

$$\tilde{\psi}' = \tfrac{1}{20+10\rho+10\rho^2+\rho^3-21\rho^4}(0, 10\rho^2(1-\rho), \rho^3(1-\rho)^3, 10\rho^2(2-\rho)(1+\rho-\rho^2), \rho^5(1-\rho), 2\rho^4(1-\rho)^2, 10\rho^4(1-\rho), 10\rho(2-\rho), 10(1-\rho)(2+\rho)).$$

Remember that $DR_T(\overline{L})/_{\mathcal{R}_{ss}(\overline{L})} = DR_{ST}(\overline{L})/_{\mathcal{R}_{ss}(\overline{L})} \cup DR_{WT}(\overline{L})/_{\mathcal{R}_{ss}(\overline{L})} = \{\widetilde{\mathcal{K}}_1, \widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}_3, \widetilde{\mathcal{K}}_5, \widetilde{\mathcal{K}}_6, \widetilde{\mathcal{K}}_8, \widetilde{\mathcal{K}}_9\}$ and $DR_V(\overline{L})/_{\mathcal{R}_{ss}(\overline{L})} = \{\widetilde{\mathcal{K}}_4, \widetilde{\mathcal{K}}_7\}$. Hence,

$$\textstyle\sum_{\widetilde{\mathcal{K}}\in DR_T(\overline{L})/_{\mathcal{R}_{ss}(\overline{L})}} \tilde{\psi}'(\widetilde{\mathcal{K}}) = \tilde{\psi}'(\widetilde{\mathcal{K}}_1) + \tilde{\psi}'(\widetilde{\mathcal{K}}_2) + \tilde{\psi}'(\widetilde{\mathcal{K}}_3) + \tilde{\psi}'(\widetilde{\mathcal{K}}_5) + \tilde{\psi}'(\widetilde{\mathcal{K}}_6) + \tilde{\psi}'(\widetilde{\mathcal{K}}_8) + \tilde{\psi}'(\widetilde{\mathcal{K}}_9) = \tfrac{20+10\rho-10\rho^2-9\rho^3-\rho^4}{20+10\rho+10\rho^2+\rho^3-21\rho^4}.$$

By Proposition 5.6, we have

$$\begin{aligned}
\tilde{\varphi}'(\widetilde{\mathcal{K}}_1) &= 0 \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = 0, \\
\tilde{\varphi}'(\widetilde{\mathcal{K}}_2) &= \tfrac{10\rho^2(1-\rho)}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}, \\
\tilde{\varphi}'(\widetilde{\mathcal{K}}_3) &= \tfrac{\rho^3(1-\rho)^3}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{\rho^3(1-\rho)^3}{20+10\rho-10\rho^2-9\rho^3-\rho^4}, \\
\tilde{\varphi}'(\widetilde{\mathcal{K}}_4) &= 0, \\
\tilde{\varphi}'(\widetilde{\mathcal{K}}_5) &= \tfrac{\rho^5(1-\rho)}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{\rho^5(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}, \\
\tilde{\varphi}'(\widetilde{\mathcal{K}}_6) &= \tfrac{2\rho^4(1-\rho)^2}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{2\rho^4(1-\rho)^2}{20+10\rho-10\rho^2-9\rho^3-\rho^4}, \\
\tilde{\varphi}'(\widetilde{\mathcal{K}}_7) &= 0, \\
\tilde{\varphi}'(\widetilde{\mathcal{K}}_8) &= \tfrac{10\rho(2-\rho)}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{10\rho(2-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}, \\
\tilde{\varphi}'(\widetilde{\mathcal{K}}_9) &= \tfrac{10(1-\rho)(2+\rho)}{20+10\rho+10\rho^2+\rho^3-21\rho^4} \cdot \tfrac{20+10\rho+10\rho^2+\rho^3-21\rho^4}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \tfrac{10(1-\rho)(2+\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}.
\end{aligned}$$

Thus, the steady-state PMF for $SMC_{\underline{\leftrightarrow}_{ss}}(\overline{L})$ is

$$\tilde{\varphi}' = \tfrac{1}{20+10\rho-10\rho^2-9\rho^3-\rho^4}(0, 10\rho^2(1-\rho), \rho^3(1-\rho)^3, 0, \rho^5(1-\rho), 2\rho^4(1-\rho)^2, 0, 10\rho(2-\rho), 10(1-\rho)(2+\rho)).$$

This coincides with the result obtained with the use of $\tilde{\psi}'^*$ and $\widetilde{SJ}'$.

Alternatively, from $TS_{\underline{\leftrightarrow}_{ss}}(\overline{L})$, we can construct the reduced quotient DTMC of $\overline{L}$, $RDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{L})$, and then calculate $\tilde{\varphi}'$ using it. By Proposition 9 from [41], it coincides with the quotient reduced DTMC of $\overline{L}$, i.e. with the quotient of $RDTMC(\overline{L})$.

Remember that $DR_{ST}(\overline{L})/_{\mathcal{R}_{ss}(\overline{L})} = \{\widetilde{\mathcal{K}}_1, \widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}_8, \widetilde{\mathcal{K}}_9\}$, $DR_{WT}(\overline{L})/_{\mathcal{R}_{ss}(\overline{L})} = \{\widetilde{\mathcal{K}}_3, \widetilde{\mathcal{K}}_5, \widetilde{\mathcal{K}}_6\}$, $DR_V(\overline{L})/_{\mathcal{R}_{ss}(\overline{L})} = \{\widetilde{\mathcal{K}}_4, \widetilde{\mathcal{K}}_7\}$. We reorder the elements of $DR(\overline{L})/_{\mathcal{R}_{ss}(\overline{L})}$,

by moving the equivalence classes of vanishing states to the first positions and those of s-tangible states to the last positions: $\widetilde{\mathcal{K}}_4, \widetilde{\mathcal{K}}_7, \widetilde{\mathcal{K}}_3, \widetilde{\mathcal{K}}_5, \widetilde{\mathcal{K}}_6, \widetilde{\mathcal{K}}_1, \widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}_8, \widetilde{\mathcal{K}}_9$.

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

$$\widetilde{\mathbf{P}}'_r = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-\rho^3 & \rho^3 & 0 & 0 \\ \frac{\rho(10-\rho)(1-\rho)}{5} & \frac{\rho^2(10-\rho)}{10} & \frac{\rho(1-\rho)^2}{10} & \frac{\rho^3}{10} & \frac{\rho^2(1-\rho)}{5} & 0 & \frac{(10-\rho)(1-\rho)^2}{10} & 0 & 0 \\ \rho^3 & 0 & 0 & 0 & 0 & 0 & \rho^2(1-\rho) & (1-\rho)(1-\rho^2) & \rho(1-\rho^2) \\ \rho^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-\rho^2 \end{pmatrix}.$$

The result of the decomposing $\widetilde{\mathbf{P}}'_r$ are the matrices

$$\widetilde{\mathbf{C}}' = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \ \widetilde{\mathbf{D}}' = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \ \widetilde{\mathbf{E}}' = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ \frac{\rho(10-\rho)(1-\rho)}{5} & \frac{\rho^2(10-\rho)}{10} \\ \rho^3 & 0 \\ \rho^2 & 0 \end{pmatrix},$$

$$\widetilde{\mathbf{F}}' = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-\rho^3 & \rho^3 & 0 & 0 \\ \frac{\rho(1-\rho)^2}{10} & \frac{\rho^3}{10} & \frac{\rho^2(1-\rho)}{5} & 0 & \frac{(10-\rho)(1-\rho)^2}{10} & 0 & 0 \\ 0 & 0 & 0 & 0 & \rho^2(1-\rho) & (1-\rho)(1-\rho^2) & \rho(1-\rho^2) \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-\rho^2 \end{pmatrix}.$$

Since $\widetilde{\mathbf{C}}'^1 = \mathbf{0}$, we have $\forall k > 0, \ \widetilde{\mathbf{C}}'^k = \mathbf{0}$, hence, $l = 0$ and there are no loops among vanishing states. Then

$$\widetilde{\mathbf{G}}' = \sum_{k=0}^{l} \widetilde{\mathbf{C}}'^l = \widetilde{\mathbf{C}}'^0 = \mathbf{I}.$$

Further, the TPM for $RDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{L})$ is

$$\widetilde{\mathbf{P}}'^{\diamond} = \widetilde{\mathbf{F}}' + \widetilde{\mathbf{E}}'\widetilde{\mathbf{G}}'\widetilde{\mathbf{D}}' = \widetilde{\mathbf{F}}' + \widetilde{\mathbf{E}}'\mathbf{I}\widetilde{\mathbf{D}}' = \widetilde{\mathbf{F}}' + \widetilde{\mathbf{E}}'\widetilde{\mathbf{D}}' =$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1-\rho^3 & \rho^3 & 0 & 0 \\ \frac{\rho(1-\rho)^2}{10} & \frac{\rho^3}{10} & \frac{\rho^2(1-\rho)}{5} & 0 & \frac{(10-\rho)(1-\rho)^2}{10} & \frac{\rho(10-\rho)(1-\rho)}{5} & \frac{\rho^2(10-\rho)}{10} \\ 0 & 0 & 0 & 0 & \rho^2(1-\rho) & 1-\rho-\rho^2+2\rho^3 & \rho(1-\rho^2) \\ 0 & 0 & 0 & 0 & 0 & \rho^2 & 1-\rho^2 \end{pmatrix}.$$

In Figure 11, the reduced quotient DTMC $RDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{L})$ is presented.

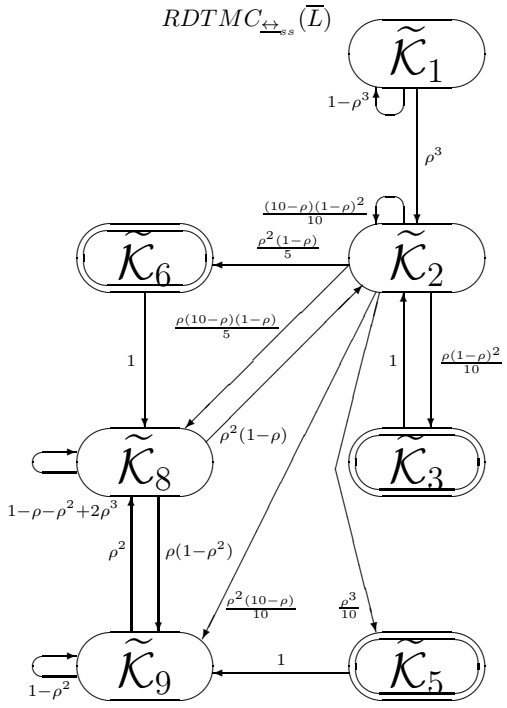


Figure 11. The reduced quotient DTMC of the abstract generalized shared memory system with maintenance.

Then the steady-state PMF for $RDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{L})$ is

$$\tilde{\psi}'^{\diamond} = \tfrac{1}{20+10\rho-10\rho^2-9\rho^3-\rho^4}(\rho^3(1-\rho)^3, \rho^5(1-\rho), 2\rho^4(1-\rho)^2, 0, 10\rho^2(1-\rho), 10\rho(2-\rho), 10(1-\rho)(2+\rho)).$$

Note that $\tilde{\psi}'^{\diamond} = (\tilde{\psi}'^{\diamond}(\widetilde{\mathcal{K}}_3), \tilde{\psi}'^{\diamond}(\widetilde{\mathcal{K}}_5), \tilde{\psi}'^{\diamond}(\widetilde{\mathcal{K}}_6), \tilde{\psi}'^{\diamond}(\widetilde{\mathcal{K}}_1), \tilde{\psi}'^{\diamond}(\widetilde{\mathcal{K}}_2), \tilde{\psi}'^{\diamond}(\widetilde{\mathcal{K}}_8), \tilde{\psi}'^{\diamond}(\widetilde{\mathcal{K}}_9))$.
By Proposition 5.8, we have

$$\begin{array}{lll}
\tilde{\varphi}'(\widetilde{\mathcal{K}}_1) = 0, & \tilde{\varphi}'(\widetilde{\mathcal{K}}_2) = \frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}, & \tilde{\varphi}'(\widetilde{\mathcal{K}}_3) = \frac{\rho^3(1-\rho)^3}{20+10\rho-10\rho^2-9\rho^3-\rho^4}, \\
\tilde{\varphi}'(\widetilde{\mathcal{K}}_4) = 0, & \tilde{\varphi}'(\widetilde{\mathcal{K}}_5) = \frac{\rho^5(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}, & \tilde{\varphi}'(\widetilde{\mathcal{K}}_6) = \frac{2\rho^4(1-\rho)^2}{20+10\rho-10\rho^2-9\rho^3-\rho^4}, \\
\tilde{\varphi}'(\widetilde{\mathcal{K}}_7) = 0, & \tilde{\varphi}'(\widetilde{\mathcal{K}}_8) = \frac{10\rho(2-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}, & \tilde{\varphi}'(\widetilde{\mathcal{K}}_9) = \frac{10(1-\rho)(2+\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}.
\end{array}$$

Thus, the steady-state PMF for $SMC_{\underline{\leftrightarrow}_{ss}}(\overline{L})$ is

$$\tilde{\varphi}' = \tfrac{1}{20+10\rho-10\rho^2-9\rho^3-\rho^4}(0, 10\rho^2(1-\rho), \rho^3(1-\rho)^3, 0, \rho^5(1-\rho), 2\rho^4(1-\rho)^2, 0, 10\rho(2-\rho), 10(1-\rho)(2+\rho)).$$

This coincides with the result obtained with the use of $\tilde{\psi}'^{*}$ and $\widetilde{SJ}'$.
We can now calculate the main performance indices.

- The average recurrence time in the state $\widetilde{\mathcal{K}}_2$, where no processor requests the memory and its maintenance is not initiated, called the *average system run-through*, is $\frac{1}{\tilde{\varphi}'_2} = \frac{20+10\rho-10\rho^2-9\rho^3-\rho^4}{10\rho^2(1-\rho)}$.
- The system is not activated only in the state $\widetilde{\mathcal{K}}_1$. Then the steady-state prob-

ability that the system is activated is $1 - \tilde{\varphi}'_1 = 1 - 0 = 1$. The common memory is available in the states $\widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}_4, \widetilde{\mathcal{K}}_7$. Then the steady-state probability that the memory is available is $\tilde{\varphi}'_2 + \tilde{\varphi}'_4 + \tilde{\varphi}'_7 = \frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4} + 0 + 0 = \frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$. The common memory is maintained only in the states $\widetilde{\mathcal{K}}_3, \widetilde{\mathcal{K}}_5, \widetilde{\mathcal{K}}_6$. Then the steady-state probability that the memory is maintained is $\tilde{\varphi}'_3 + \tilde{\varphi}'_5 + \tilde{\varphi}'_6 = \frac{\rho^3(1-\rho)^3}{20+10\rho-10\rho^2-9\rho^3-\rho^4} + \frac{\rho^5(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4} + \frac{2\rho^4(1-\rho)^2}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \frac{\rho^3(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$. The steady-state probability that the memory is used (i.e. neither available nor maintained), the *shared memory utilization*, is $1 - \frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4} - \frac{\rho^3(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4} = \frac{10(2+\rho-2\rho^2)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$.

- After activation of the system, we leave the state $\widetilde{\mathcal{K}}_1$ for ever, and the common memory is either requested or allocated or maintained in every remaining state, with exception of $\widetilde{\mathcal{K}}_2$. Thus, the *rate with which the necessity (also for maintenance) of shared memory emerges* coincides with the rate of leaving $\widetilde{\mathcal{K}}_2$, calculated as $\frac{\tilde{\varphi}'_2}{\widetilde{SJ}'_2} = \frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4} \cdot \frac{\rho(21-12\rho+\rho^2)}{10} = \frac{\rho^3(1-\rho)(21-12\rho+\rho^2)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$.
- The parallel common memory request of two processors $\{\{r\},\{r\}\}$ is possible from the state $\widetilde{\mathcal{K}}_2$. In this state, the request probability is the sum of the execution probabilities for all multisets of multiactions containing $\{r\}$ twice. The *steady-state probability of the shared memory request from two processors* is $\tilde{\varphi}'_2 \sum_{\{A,\widetilde{\mathcal{K}}|\{\{r\},\{r\}\}\subseteq A,\ \widetilde{\mathcal{K}}_2 \stackrel{A}{\to} \widetilde{\mathcal{K}}\}} PM_A(\widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}) = \frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4} \left(\frac{\rho^2(10-\rho)}{10} + \frac{\rho^3}{10}\right) = \frac{10\rho^4(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$.
- The common memory request of a processor $\{r\}$ is possible from the states $\widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}_8$. In each of them, the request probability is the sum of the execution probabilities for all multisets of multiactions containing $\{r\}$. The *steady-state probability of the shared memory request from a processor* is $\tilde{\varphi}'_2 \sum_{\{A,\widetilde{\mathcal{K}}|\{r\}\in A,\ \widetilde{\mathcal{K}}_2 \stackrel{A}{\to} \widetilde{\mathcal{K}}\}} PM_A(\widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}) + \tilde{\varphi}'_8 \sum_{\{A,\widetilde{\mathcal{K}}|\{r\}\in A,\ \widetilde{\mathcal{K}}_8 \stackrel{A}{\to} \widetilde{\mathcal{K}}\}} PM_A(\widetilde{\mathcal{K}}_8, \widetilde{\mathcal{K}}) = \frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4} \cdot \left(\frac{\rho(10-\rho)(1-\rho)}{5} + \frac{\rho^2(1-\rho)}{5} + \frac{\rho^2(10-\rho)}{10} + \frac{\rho^3}{10}\right) + \frac{10\rho(2-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}(\rho(1-\rho^2)+\rho^3) = \frac{10\rho^2(2-\rho)(1+\rho-\rho^2)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$.

The performance indices are the same for the 'complete' and the 'quotient' abstract generalized shared memory systems with maintenance. The coincidence of the first, second and third performance indices obviously illustrates the results of Proposition 6.1 and Proposition 6.4 (both modified for $\mathcal{R}_{\mathcal{L}ss}(\overline{L})$). The coincidence of the fourth performance index is due to Theorem 6.3 (modified for $\mathcal{R}_{\mathcal{L}ss}(\overline{L})$): one should just apply its result to the derived step traces $\{\{r\},\{r\}\}$ and $\{\{r\},\{r\},\{c\}\}$ of the expression $\overline{L}$ and itself, and then sum the left and right parts of the two resulting equalities. The coincidence of the fifth performance index is due to Theorem 6.3 (modified for $\mathcal{R}_{\mathcal{L}ss}(\overline{L})$): one should just apply its result to the derived step traces $\{\{r\}\}$, $\{\{r\},\{c\}\}$, $\{\{r\},\{r\}\}$, $\{\{r\},\{r\},\{c\}\}$, $\{\{r\},\{m\}\}$ of the expression $\overline{L}$ and itself, and then sum the left and right parts of the five resulting equalities.

Let us consider the effect of quantitative changes of the parameter ρ upon performance of the 'quotient' abstract generalized shared memory system with maintenance in its steady state. Remember that $\rho \in (0;1)$ is the probability of every stochastic multiaction in the specification of the system. The closer is ρ to 0, the less is the probability to execute some activities at every discrete time tick, hence, the system will most probably *stand idle*. The closer is ρ to 1, the greater is the probability to execute some

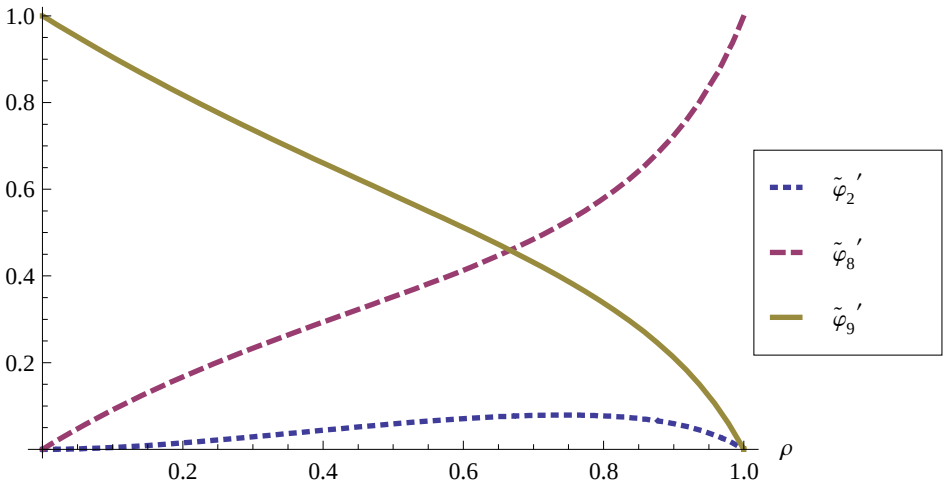


Figure 12. Steady-state probabilities $\tilde{\varphi}'_2$, $\tilde{\varphi}'_8$, $\tilde{\varphi}'_9$ (large probability masses) as functions of the parameter ρ.

activities at every discrete time tick, hence, the system will most probably *operate*.

Since $\tilde{\varphi}'_1 = \tilde{\varphi}'_4 = \tilde{\varphi}'_7 = 0$, only $\tilde{\varphi}'_2 = \frac{10\rho^2(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$, $\tilde{\varphi}'_3 = \frac{\rho^3(1-\rho)^3}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$, $\tilde{\varphi}'_5 = \frac{\rho^5(1-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$, $\tilde{\varphi}'_6 = \frac{2\rho^4(1-\rho)^2}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$, $\tilde{\varphi}'_8 = \frac{10\rho(2-\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$, $\tilde{\varphi}'_9 = \frac{10(1-\rho)(2+\rho)}{20+10\rho-10\rho^2-9\rho^3-\rho^4}$ depend on ρ. In Figure 12, the plots of $\tilde{\varphi}'_2$, $\tilde{\varphi}'_8$, $\tilde{\varphi}'_9$ (large probability masses) as functions of ρ are depicted. In Figure 13, the plots of $\tilde{\varphi}'_3$, $\tilde{\varphi}'_5$, $\tilde{\varphi}'_6$ (small probability masses) as functions of ρ are drawn. We do not allow $\rho = 0$ or $\rho = 1$.

Then $\tilde{\varphi}'_2$, $\tilde{\varphi}'_3$, $\tilde{\varphi}'_5$, $\tilde{\varphi}'_6$, $\tilde{\varphi}'_8$ tend to 0 and $\tilde{\varphi}'_9$ tends to 1 when ρ approaches 0. Thus, when ρ is closer to 0, the probability that the memory is allocated to a processor and requested by another processor increases, implying *more unsatisfied memory requests*.

Next, $\tilde{\varphi}'_2$, $\tilde{\varphi}'_3$, $\tilde{\varphi}'_5$, $\tilde{\varphi}'_6$, $\tilde{\varphi}'_9$ tend to 0 and $\tilde{\varphi}'_8$ tends to 1 when ρ approaches 1. Thus, when ρ is closer to 1, the probability that the memory is allocated to a processor (and not requested by another one) increases, implying *less unsatisfied memory requests*.

The maximal value 0.0792 of $\tilde{\varphi}'_2$ is reached when $\rho \approx 0.7427$. Then the probability that the system is activated and the memory is not requested and its maintenance is not initiated is maximal, i.e. the *maximal shared memory availability* is about 8%.

The maximal value 0.0007 of $\tilde{\varphi}'_3$ is reached when $\rho \approx 0.5158$. Then the probability that the memory maintenance is initiated is maximal, i.e. the *maximal shared memory maintenance necessity* is about 0.1%.

The maximal value 0.0044 of $\tilde{\varphi}'_5$ is reached when $\rho \approx 0.8724$. then the probability that the memory maintenance is initiated and the memory is requested by two processors is maximal, i.e. the *maximal double (parallel) demand of shared memory during its maintenance* is about 0.4%.

The maximal value 0.0023 of $\tilde{\varphi}'_6$ is reached when $\rho \approx 0.7015$. Then the probability that the memory maintenance is initiated and the memory is requested by a (single) processor is maximal, i.e. the *maximal single demand of shared memory during its maintenance* is about 0.2%.

In Figure 14, the plot of the average system run-through, calculated as $\frac{1}{\tilde{\varphi}'_2}$, as a function of ρ is depicted. The run-through tends to ∞ when ρ approaches 0 or 1. Its minimal value 12.6259 is reached when $\rho \approx 0.7427$. To speed up operation of the system, one should take the parameter ρ closer to 0.7427.

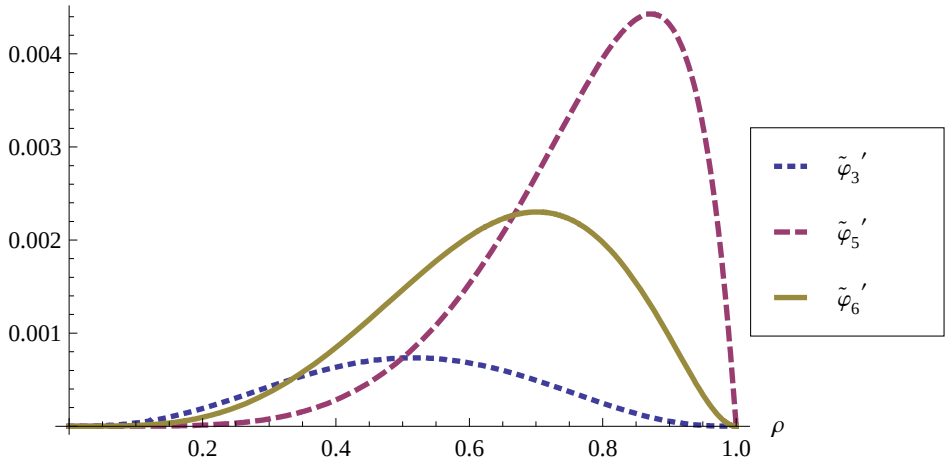


Figure 13. Steady-state probabilities $\tilde{\varphi}_3'$, $\tilde{\varphi}_5'$, $\tilde{\varphi}_6'$ (small probability masses) as functions of the parameter ρ.

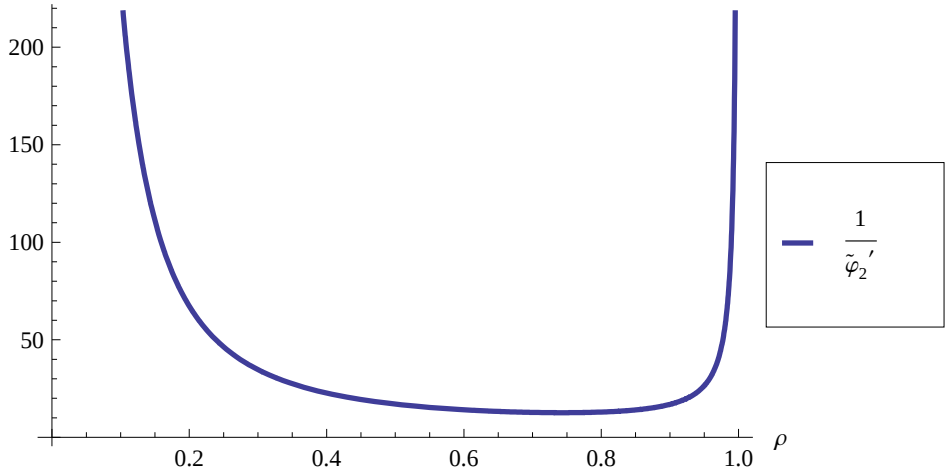


Figure 14. Average system run-through $\frac{1}{\tilde{\varphi}_2'}$ as a function of the parameter ρ.

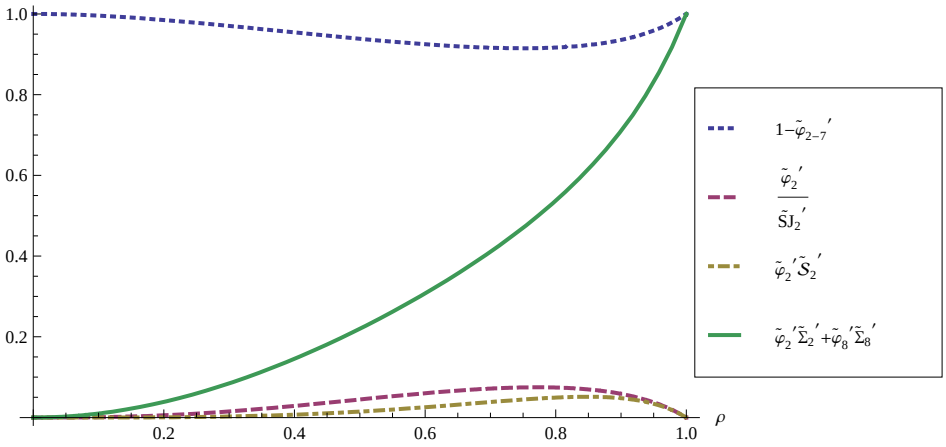


Figure 15. Some performance indices as functions of the parameter ρ.

The first curve in Figure 15 represents the shared memory utilization, calculated as $1-\tilde{\varphi}'_{2-7}$, where $\tilde{\varphi}'_{2-7} = \tilde{\varphi}'_2+\tilde{\varphi}'_3+\tilde{\varphi}'_4+\tilde{\varphi}'_5+\tilde{\varphi}'_6+\tilde{\varphi}'_7$, as a function of ρ. The utilization tends to 1 when ρ approaches 0 or 1. The minimal value 0.9149 of the utilization is reached when $\rho \approx 0.7494$. Thus, the *minimal shared memory utilization* is about 91%. To increase the utilization, one should take the parameter ρ closer to 0 or 1.

The second curve in Figure 15 represents the rate with which the necessity of shared memory emerges, calculated as $\frac{\tilde{\varphi}'_2}{\widetilde{SJ}_2}$, as a function of ρ. The rate tends to 0 when ρ approaches 0 or 1. The maximal value 0.0749 of the rate is reached when $\rho \approx 0.7723$. Thus, the *maximal rate with which the necessity of shared memory emerges* is about $\frac{1}{13}$. To decrease the rate, one should take the parameter ρ closer to 0 or 1.

The third curve in Figure 15 represents the steady-state probability of the shared memory request from two processors, calculated as $\tilde{\varphi}'_2\widetilde{\mathcal{S}}'_2$, where $\widetilde{\mathcal{S}}'_2 = \sum_{\{A,\widetilde{\mathcal{K}}|\{\{r\},\{r\}\}\subseteq A,\ \widetilde{\mathcal{K}}_2 \stackrel{A}{\to} \widetilde{\mathcal{K}}\}} PM_A(\widetilde{\mathcal{K}}_2,\widetilde{\mathcal{K}})$, as function of ρ. The probability tends to 0 when ρ approaches 0 or 1. Its maximal value 0.0514 is reached when $\rho \approx 0.8486$. To decrease the probability, one should take the parameter ρ closer to 0 or 1.

The fourth curve in Figure 15 represents the steady-state probability of the shared memory request from a processor, calculated as $\tilde{\varphi}'_2\widetilde{\Sigma}'_2 + \tilde{\varphi}'_8\widetilde{\Sigma}'_8$, where $\widetilde{\Sigma}'_i = \sum_{\{A,\widetilde{\mathcal{K}}|\{r\}\in A,\ \widetilde{\mathcal{K}}_i \stackrel{A}{\to} \widetilde{\mathcal{K}}\}} PM_A(\widetilde{\mathcal{K}}_i,\widetilde{\mathcal{K}})$, $i \in \{2,8\}$, as a function of ρ. The probability tends to 0 when ρ approaches 0 and to 1 when ρ approaches 1. To increase the probability, one should take the parameter ρ closer to 1.

8. Discussion

We now sum up the work presented, explain its specifics and outline planned research.

8.1. Results obtained

In this paper, we have considered dtsdPBC, an extension with discrete stochastic and deterministic time of Petri box calculus (PBC) [7–9]. Stochastic process algebra dtsdPBC has a parallel step operational semantics, based on labeled probabilistic tran-

sition systems, and a Petri net denotational semantics in terms of dtsd-boxes, a special subclass of labeled discrete time stochastic and deterministic Petri nets (LDTSDPNs) [38,39]. Step stochastic bisimulation equivalence of the process expressions is used to reduce with the quotienting their transition systems and (semi-, discrete time and reduced discrete time) Markov chains (SMCs, DTMCs and RDTMCs). That equivalence guarantees identity of the steady-state probabilities, sojourn time averages and variances in the equivalence classes. Hence, the equivalence preserves the stationary performance measures and can be used for minimization of the state space.

The properties of the mentioned equivalence permit to provide dtsdPBC with a method of modeling, quotient reduction and simplified performance evaluation of concurrent stochastic systems that maintains the semantic parallelism, which stems from simultaneous executions. We have presented a case study of a generalization of the shared memory system with maintenance, by allowing for variable probabilities and weights in its specification. The variable generalized probability has been interpreted as a parameter of the performance index functions. The influence of the parameter value to the system's performance has been analyzed with the goal of optimization.

8.2. Originality of the model

The merit 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. As argued in [53], (stochastic) Petri nets (PNs) represent the systems structure more concisely and can be an intermediate formalism for their more intuitive translation into 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.

The denotational semantics of dtsdPBC is constructed via dtsd-boxes, a safe (at most one token in each place at every reachable marking) and interface-featured (the place and transition labeling for composition) version of LDTSDPNs that themselves are based on discrete time stochastic PNs (DTSPNs) [31,32], extended with transition labeling and deterministic transitions. The key idea is to interpret the waiting (positively delayed) transitions with the timer values one in LDTSDPNs as the (stochastic) transitions with the conditional probability 1 in DTSPNs. Then the waiting transitions with the timer values greater than one are ignored: when enabled, they are executed with the probability 0 at the next moment. Since the enabled waiting transitions with the timer values one in LDTSDPNs cannot be delayed, all such non-conflicting transitions should fire at the next moment. Hence, only *maximal* non-empty sets of them are fired, with the overall probability 1, collected over all the alternative transition sets.

In [54], an extension of Timed Petri nets (TPNs) [14], called TPNs with multichannel transitions (MCTPNs), were defined that allow simultaneous starting of (possibly the same) transitions and adapt the *multiple server* policy [55]. MCTPNs were successfully applied in the automated manufacturing scheduling and for visualization of the production processes in dispatching tool systems within machine-building enterprizes. LDTSDPNs may be non-safe, but they do not have self-concurrency (firing transitions in parallel to themselves), to avoid technical difficulties with calculating (normalized)

probabilities for multisets of transitions. Since dtsd-boxes are safe LDTSDPNs, the former have no self-concurrency as well, hence, the *single server* policy [55] is applied in the both models. In contrast to TPNs and MCTPNs, the enabled transitions in LDTSDPNs do not consume tokens from their input places, thus allowing the enabled stochastic transitions to preempt (interrupt) the enabled waiting ones that cannot be executed at the next moment. LDTSDPNs also have immediate (zero delayed) transitions, in order to model logical conditions, probabilistic branching, instantaneous probabilistic choices and activities with negligible durations. LDTSDPNs are similar to *synchronous* MCTPNs [54] in that the maximal (multi)set of enabled (waiting) transitions is started (fired) at each time step, and the both adapt the *strong time* policy [56].

dtsdPBC 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. These include Dirac distribution of the positive deterministic sojourn time, which is then splitted into one time units and allocated with the consecutive process states. In addition, dtsdPBC can model the mentioned systems featuring very scattered activity delays, or even more complex systems with immediate probabilistic choice or urgency. dtsdPBC is well suited for the discrete time applications, whose discrete states change with a global time tick, such as business processes, neural and transportation networks, computer and communication systems, timed web services [57], as well as for those, whose distributed architecture or the concurrency level should be preserved while modeling and analysis, such as genetic regulatory and cellular signalling networks (featuring maximal parallelism) in biology [58,59]. dtsdPBC can also model and analyze parallel systems with fixed durations of the typical activities (loading, processing, transfer, repair, low-level events, message delivery) and stochastic durations of the randomly occurring activities (arrival, departure, failure, packet loss, message collision), including industrial, manufacturing, queueing, computing and network systems.

8.3. Research perspectives

Future work consists in constructing a congruence relation for dtsdPBC, i.e. the equivalence that withstands application of all operations of the algebra. A possible candidate is a stronger version of the equivalence with respect to transition systems, with two extra transitions skip and redo, like in sPBC [21]. Moreover, recursion operation could be added to dtsdPBC to increase specification power of the algebra.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Hoare CAR. Communicating sequential processes. Prentice-Hall, London, UK; 1985.

[2] Bergstra JA, Klop JW. Algebra of communicating processes with abstraction. Theor Comput Sci. 1985;37:77–121.

[3] Milner RAJ. Communication and concurrency. Prentice-Hall, NJ, USA; 1989.

[4] Hermanns H, Rettelbach M. Syntax, semantics, equivalences and axioms for MTIPP. In: Herzog U, Rettelbach M, editors. Proc. 2^{nd} Int. Workshop on Process Algebra and

Performance Modelling (PAPM) 1994; (Arbeitsberichte des IMMD; Vol. 27). Universität Erlangen-Nürnberg, Germany; 1994. p. 71–88.
[5] Hillston J. A compositional approach to performance modelling. Cambridge University Press, Cambridge, UK; 1996.
[6] Bernardo M, Gorrieri R. A tutorial on EMPA: a theory of concurrent processes with nondeterminism, priorities, probabilities and time. Theor Comput Sci. 1998;202:1–54.
[7] Best E, Devillers R, Hall JG. The box calculus: a new causal algebra with multi-label communication. In: Rozenberg G, editor. Advances in Petri Nets (APN) 1992; (Lect. Notes Comp. Sci.; Vol. 609). Springer; 1992. p. 21–69.
[8] Best E, Koutny M. A refined view of the box algebra. In: Michelis GD, Diaz M, editors. Proc. 16^{th} Int. Conf. on Application and Theory of Petri Nets (ICATPN) 1995; (Lect. Notes Comp. Sci.; Vol. 935). Springer; 1995. p. 1–20.
[9] Best E, Devillers R, Koutny M. Petri net algebra. Springer; 2001. EATCS Monographs on Theor. Comput. Sci.
[10] Koutny M. A compositional model of time Petri nets. In: Nielsen M, Simpson D, editors. Proc. 21^{st} Int. Conf. on Application and Theory of Petri Nets (ICATPN) 2000; (Lect. Notes Comp. Sci.; Vol. 1825). Springer; 2000. p. 303–322.
[11] Merlin PM, Farber DJ. Recoverability of communication protocols: implications of a theoretical study. IEEE Trans Communications. 1976;24:1036–1043.
[12] Marroquín A O, de Frutos E D. TPBC: timed Petri box calculus. Madrid, Spain: Departamento de Sistemas Infofmáticos y Programación, Universidad Complutense de Madrid; 2000. Technical report. In Spanish.
[13] Marroquín A O, de Frutos E D. Extending the Petri box calculus with time. In: Colom JM, Koutny M, editors. Proc. 22^{nd} Int. Conf. on Applications and Theory of Petri Nets (ICATPN) 2001; (Lect. Notes Comp. Sci.; Vol. 2075). Springer; 2001. p. 303–322.
[14] Ramchandani C. Perfomance evaluation of asynchronous concurrent systems by timed Petri nets [dissertation]. Cambridge, USA: Massachusetts Institute of Technology; 1973.
[15] Niaouris A. An algebra of Petri nets with arc-based time restrictions. In: Liu Z, Araki K, editors. Proc. 1^{st} Int. Colloquium on Theoretical Aspects of Computing (ICTAC) 2004; (Lect. Notes Comp. Sci.; Vol. 3407). Springer; 2005. p. 447–462.
[16] Niaouris A, Koutny M. An algebra of timed-arc Petri nets. Newcastle upon Tyne, UK: School of Computer Science, University of Newcastle upon Tyne; 2005. Technical Report CS-TR-895. Available from: http://www.cs.ncl.ac.uk/publications/trs/papers/895.pdf.
[17] Bolognesi T, Lucidi F, Trigila S. From timed Petri nets to timed LOTOS. In: Logrippo L, Probert RL, Ural H, editors. Proc. IFIP WG6.1 10^{th} Int. Symposium on Protocol Specification, Testing and Verification (PSTV) 1990. Elsevier Science Publishers (North-Holland), Amsterdam, The Netherlands; 1990. p. 395–408.
[18] Hanish HM. Analysis of place/transition nets with timed-arcs and its application to batch process control. In: Marsan MA, editor. Proc. 14^{th} Int. Conf. on Application and Theory of Petri Nets (ICATPN) 1993; (Lect. Notes Comp. Sci.; Vol. 691). Springer; 1993. p. 282–299.
[19] Macià S H, Valero R V, de Frutos E D. sPBC: a Markovian extension of finite Petri box calculus. In: Proc. 9^{th} IEEE Int. Workshop on Petri Nets and Performance Models (PNPM) 2001. IEEE Computer Society Press; 2001. p. 207–216.
[20] Macià S H, Valero R V, Cazorla L DC, et al. Introducing the iteration in sPBC. In: de Frutos E D, Núñez G M, editors. Proc. 24^{th} Int. Conf. on Formal Techniques for Networked and Distributed Systems (FORTE) 2004; (Lect. Notes Comp. Sci.; Vol. 3235). Springer; 2004. p. 292–308.
[21] Macià S H, Valero R V, Cuartero G F, et al. A congruence relation for sPBC. Form Methods Syst Des. 2008;32:85–128.
[22] Marsan MA. Stochastic Petri nets: an elementary introduction. In: Rozenberg G, editor. Advances in Petri Nets (APN) 1989; (Lect. Notes Comp. Sci.; Vol. 424). Springer; 1990. p. 1–29.
[23] Balbo G. Introduction to stochastic Petri nets. In: Brinksma E, Hermanns H, Katoen JP,

editors. Proc. 1^{st} EEF/Euro Summer School of Trends in Comp. Sci. 2000; (Lect. Notes Comp. Sci.; Vol. 2090). Springer; 2001. p. 84–155.
[24] Macià S H, Valero R V, Cuartero G F, et al. sPBC: a Markovian extension of Petri box calculus with immediate multiactions. Fundam Inform. 2008;87:367–406.
[25] Macià S H, Valero R V, Cuartero G F, et al. Modelling a video conference system with sPBC. Appl Math Inf Sci. 2016;10:475–493.
[26] Balbo G. Introduction to generalized stochastic Petri nets. In: Bernardo M, Hillston J, editors. Formal Methods for Performance Evaluation. Proc. 7^{th} Int. School on Formal Methods for the Design of Computer, Communication, and Software Systems (SFM) 2007; (Lect. Notes Comp. Sci.; Vol. 4486). Springer; 2007. p. 83–131.
[27] Tarasyuk IV. Discrete time stochastic Petri box calculus. Oldenburg, Germany: Carl von Ossietzky Universität Oldenburg; 2005. Berichte aus dem Department für Informatik 3/05. Available from: http://itar.iis.nsk.su/files/itar/pages/dtspbcib_cov.pdf.
[28] Tarasyuk IV. Iteration in discrete time stochastic Petri box calculus. Bull Nov Comp Center, Comp Science, IIS Special Issue. 2006;24:129–148.
[29] Tarasyuk IV. Stochastic Petri box calculus with discrete time. Fundam Inform. 2007; 76:189–218.
[30] Tarasyuk IV. Equivalence relations for modular performance evaluation in dtsPBC. Math Struct Comp Sci. 2014;24:78–154 (e240103).
[31] Molloy MK. On the integration of the throughput and delay measures in distributed processing models [dissertation]. Los Angeles, CA, USA: University of California; 1981.
[32] Molloy MK. Discrete time stochastic Petri nets. IEEE Trans Software Eng. 1985;11:417–423.
[33] Tarasyuk IV, Macià S H, Valero R V. Discrete time stochastic Petri box calculus with immediate multiactions. Albacete, Spain: Department of Computer Systems, High School of Computer Science Engineering, University of Castilla - La Mancha; 2010. Technical Report DIAB-10-03-1.
[34] Tarasyuk IV, Macià S H, Valero R V. Discrete time stochastic Petri box calculus with immediate multiactions dtsiPBC. In: Bradley J, Heljanko K, Knottenbelt W, et al., editors. Proc. 6^{th} Int. Workshop on Practical Applications of Stochastic Modelling (PASM) 2012 and 11^{th} Int. Workshop on Parallel and Distributed Methods in Verification (PDMC) 2012; (Electronic Notes in Theor. Comp. Sci.; Vol. 296). Elsevier; 2013. p. 229–252.
[35] Tarasyuk IV, Macià S H, Valero R V. Performance analysis of concurrent systems in algebra dtsiPBC. Programming and Computer Software. 2014;40:229–249.
[36] Tarasyuk IV, Macià S H, Valero R V. Stochastic process reduction for performance evaluation in dtsiPBC. Siberian Electronic Mathematical Reports. 2015;12:513–551.
[37] Tarasyuk IV, Macià S H, Valero R V. Stochastic equivalence for performance analysis of concurrent systems in dtsiPBC. Siberian Electronic Mathematical Reports. 2018;15:1743–1812.
[38] Tarasyuk IV. Discrete time stochastic and deterministic Petri box calculus. Ithaca, NY, USA: Computing Research Repository, Cornell University Library; 2019. CoRR abs/1905.00456.
[39] Tarasyuk IV. Discrete time stochastic and deterministic Petri box calculus dtsdPBC. Siberian Electronic Mathematical Reports. 2020;17:1598–1679.
[40] Tarasyuk IV. Performance evaluation in stochastic process algebra dtsdPBC. Siberian Electronic Mathematical Reports. 2021;18:1105–1145.
[41] Tarasyuk IV. Performance preserving equivalence for stochastic process algebra dtsdPBC. Siberian Electronic Mathematical Reports. 2023;20:646–699.
[42] Tarasyuk IV. Embedding and elimination for performance analysis in stochastic process algebra dtsdPBC. International Journal of Parallel, Emergent and Distributed Systems. 2024;39:619–652.
[43] Zijal R, German R. A new approach to discrete time stochastic Petri nets. In: Cohen G, Quadrat JP, editors. Proc. 11^{th} Int. Conf. on Analysis and Optimization of Systems, Discrete Event Systems (DES) 1994; (Lecture Notes in Control and Information Sciences;

Vol. 199). Springer; 1994. p. 198–204.
[44] Zijal R. Discrete time deterministic and stochastic Petri nets. In: Hommel G, editor. Proc. Int. Workshop on Quality of Communication-Based Systems 1994; Technical University of Berlin, Germany. Kluwer Academic Publishers; 1995. p. 123–136.
[45] Zijal R. Analysis of discrete time deterministic and stochastic Petri nets [dissertation]. Berlin, Germany: Technical University of Berlin; 1997.
[46] Marsan MA, Balbo G, Conte G, et al. Modelling with generalized stochastic Petri nets. John Wiley and Sons; 1995. Wiley Series in Parallel Computing.
[47] van der Aalst WMP, van Hee KM, Reijers HA. Analysis of discrete-time stochastic Petri nets. Statistica Neerlandica. 2000;54:237–255.
[48] Jou CC, Smolka SA. Equivalences, congruences and complete axiomatizations for probabilistic processes. In: Baeten JCM, Klop JW, editors. Proc. 1^{st} Int. Conf. on Theories of Concurrency: Unification and Extension (CONCUR) 1990; (Lect. Notes Comp. Sci.; Vol. 458). Springer; 1990. p. 367–383.
[49] Larsen KG, Skou A. Bisimulation through probabilistic testing. Inform Comput. 1991; 94:1–28.
[50] Bernardo M. A survey of Markovian behavioral equivalences. In: Bernardo M, Hillston J, editors. Formal Methods for Performance Evaluation. Proc. 7^{th} Int. School on Formal Methods for the Design of Computer, Communication, and Software Systems (SFM) 2007; (Lect. Notes Comp. Sci.; Vol. 4486). Springer; 2007. p. 180–219.
[51] Bernardo M. Non-bisimulation-based Markovian behavioral equivalences. J Log Algebr Program. 2007;72:3–49.
[52] Kulkarni VG. Modeling and analysis of stochastic systems. 2^{nd} ed. Chapman and Hall / CRC Press; 2010. Texts in Statistical Science.
[53] Trivedi KS. Probability and statistics with reliability, queuing, and computer science applications. 2^{nd} ed. John Wiley and Sons, Hoboken, NJ, USA; 2016.
[54] Zaitsev DA, Sleptsov AI. State equations and equivalent transformations for timed Petri nets. Cybern Syst Anal. 1997;33:659–672.
[55] Boucheneb H, Lime D, Roux OH. On multi-enabledness in time Petri nets. In: Colom JM, Desel J, editors. Proc. 34^{th} Int. Conf. on Application and Theory of Petri Nets and Concurrency (ICATPN) 2013; (Lect. Notes Comp. Sci.; Vol. 7927). Springer; 2013. p. 130–149.
[56] Boyer M, Roux OH. Comparison of the expressiveness of arc, place and tranition time Petri nets. In: Kleijn J, Yakovlev A, editors. Proc. 28^{th} Int. Conf. on Application and Theory of Petri Nets and Concurrency (ICATPN) 2007; (Lect. Notes Comp. Sci.; Vol. 4546). Springer; 2007. p. 63–82.
[57] Valero R V, Cambronero P ME. Using unified modelling language to model the publish/subscribe paradigm in the context of timed Web services with distributed resources. Mathematical and Computer Modelling of Dynamical Systems. 2017;23:570–594.
[58] Bonzanni N, Feenstra KA, Fokkink WJ, et al. What can formal methods bring to systems biology? In: Cavalcanti A, Dams DR, editors. Proc. 2^{nd} World Congress on Formal Methods (FM) 2009; (Lect. Notes Comp. Sci.; Vol. 5850). Springer; 2009. p. 16–22.
[59] Bartocci E, Lió P. Computational modeling, formal analysis, and tools for systems biology. PLoS Comput Biol. 2015;12:1–22 (e1004591).