S©MR ISSN 1813-3304

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ

Siberian Electronic Mathematical Reports

http://semr.math.nsc.ru

Том 20, № 2, стр. 646–699 (2023) УДК 004.423.4, 519.217.2, 519.681

DOI 10.33048/semi.2023.20.039 MSC 18C10, 60J10, 60K15, 68Q55, 68Q85

PERFORMANCE PRESERVING EQUIVALENCE FOR STOCHASTIC PROCESS ALGEBRA DTSDPBC

I.V. TARASYUK

Abstract. Petri box calculus (PBC) of E. Best, R. Devillers, J.G. Hall and M. Koutny is a well-known algebra of parallel processes with a Petri net semantics. Discrete time stochastic and deterministic PBC (dtsdPBC) of the author extends PBC with discrete time stochastic and deterministic delays. dtsdPBC has a step operational semantics via labeled probabilistic transition systems and a Petri net denotational semantics via dtsd-boxes, a subclass of labeled discrete time stochastic and deterministic Petri nets (LDTSDPNs). To evaluate performance in dtsdPBC, the underlying semi-Markov chains (SMCs) and (reduced) discrete time Markov chains (DTMCs and RDTMCs) of the process expressions are analyzed. Step stochastic bisimulation equivalence is used in dtsdPBC as to compare the qualitative and quantitative behaviour, as to establish consistency of the operational and denotational semantics.

We demonstrate how to apply step stochastic bisimulation equivalence of the process expressions for quotienting their transition systems, SMCs, DTMCs and RDTMCs while preserving the stationary behaviour and residence time properties. We also prove that the quotient behavioural structures (transition systems, reachability graphs and SMCs) of the process expressions and their dtsd-boxes are isomorphic. Since the equivalence guarantees identity of the functional and performance characteristics in the equivalence classes, it can be used to simplify performance analysis within dtsdPBC due to the quotient minimization of the state space.

Keywords: Petri box calculus, discrete time, stochastic and deterministic delays, transition system, operational semantics, dtsd-box, denotational semantics, Markov chain, performance, stochastic bisimulation, quotient.

Tarasyuk, I.V., Performance preserving equivalence for stochastic process algebra dtsdPBC.

© 2023 Tarasyuk I.V.

Received May, 16, 2022, published July, 31, 2023.

1. Introduction

Process calculi, like CSP [47], ACP [8] and CCS [69] are well-known formal models for specification of computing systems and analysis of their behaviour. In such process algebras (PAs), formulas describe processes, and verification of the functionality properties of their behaviour 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 [45], PEPA [46] and EMPA [18]. 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) [21, 23, 22] 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 [51]. 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 [68] and called time Petri boxes (ct-boxes).

Another time enrichment of PBC, called Timed Petri box calculus (TPBC), was defined in [64, 65], 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 [76] and called Timed Petri boxes (T-boxes).

The third time extension of PBC, called arc time Petri box calculus (atPBC), was constructed in [72, 73], 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 [24, 43], where time restrictions are associated with the arcs, called arc time Petri boxes (at-boxes). tPBC, TPBC and atPBC, all adapt the discrete time approach, 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 [60, 56, 57]. 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 [66, 4] and called stochastic Petri boxes (s-boxes). In [57], a number of new equivalence relations were proposed for regular terms of sPBC to choose later a suitable candidate for a congruence.

sPBC was enriched with immediate multiactions having zero delay in [58, 59]. 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 [66, 4, 5] and called generalized stochastic Petri boxes (gs-boxes).

In [81, 82, 83, 85], 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 [70, 71] and called discrete time stochastic Petri boxes (dts-boxes). A variety of stochastic equivalences were proposed to identify stochastic processes with similar behaviour which are differentiated by the semantic equivalence. The interrelations of all the introduced equivalences were studied.

In [89, 90, 91, 92, 93], 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. Step stochastic bisimulation equivalence of the expressions was defined to compare and reduce their transition systems and Markov chains, as well as to identify the stationary behaviour.

In [86, 87], 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 [100, 98, 99], 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. **Equivalence relations.** A notion of equivalence is very important in theory of computing systems. Equivalences are applied both to compare behaviour of systems and reduce their structure. There is a wide diversity of behavioural equivalences, and their interrelations are well explored in the literature. The best-known and widely used one is bisimulation. Typically, the mentioned equivalences take into account only functional (qualitative) but not performance (quantitative) aspects. Additionally, the equivalences are usually interleaving ones, i.e. they interpret concurrency as sequential nondeterminism. Interleaving equivalences permit to imitate parallel execution of actions via all possible occurrence sequences (interleavings) of them. Step equivalences require instead simulating such a parallel execution by simultaneous occurrence (step) of all the involved actions.

To respect quantitative features of behaviour, probabilistic equivalences have additional requirement on execution probabilities. Two equivalent processes must be able to execute the same sequences of actions, and for every such sequence, its execution probabilities within both processes should coincide. In case of probabilistic bisimulation equivalence, the states from which similar future behaviours start are grouped into equivalence classes that form elements of the aggregated state space. From every two bisimilar states, the same actions can be executed, and the subsequent states resulting from execution of an action belong to the same equivalence class. In addition, for both states, the cumulative probabilities to move to the same equivalence class by executing the same action coincide. A different kind of quantitative relations is called Markovian equivalences, which take rate (the parameter of exponential distribution that governs time delays) instead of probability. The probabilistic equivalences can be seen as discrete time analogues of the Markovian ones, since the latter are defined as the continuous time relations.

Interleaving probabilistic weak trace equivalence was introduced in [35] on labeled probabilistic transition systems. Interleaving probabilistic strong bisimulation equivalence was proposed in [54] on the same model. Interleaving probabilistic equivalences were defined for probabilistic processes in [50, 41]. Interleaving Markovian strong bisimulation equivalence was constructed in [45] for MTIPP, in [46] for PEPA and in [18, 17, 9] for EMPA. Several variants of interleaving Markovian weak bisimulation equivalence were considered in [29] on Markovian process algebras, in [31] on labeled CTSPNs and in [32] on labeled GSPNs. In [33, 34, 84], interleaving and step probabilistic trace and bisimulation equivalences that abstract from silent actions were defined on labeled DTSPNs (LDTSPNs) with invisible transitions, including the back and back-forth variants of the considered bisimulation relations. In [14, 15], interleaving probabilistic and Markovian trace, testing and bisimulation equivalences on the respective sequential probabilistic (PPC) and Markovian (MPC) process calculi were logically characterized. In [10, 11, 12], a comparison of interleaving Markovian trace, test, strong and weak bisimulation equivalences was carried out on sequential (SMPC or MPC) and concurrent (CMPC) Markovian process calculi. In [36], interleaving strong and branching probabilistic bisimulation equivalences were defined on the calculus of Interactive Probabilistic Chains (IPC).

Next, in [19, 20, 13], a lot of probabilistic and Markovian trace, testing and bisimulation equivalences were investigated on Uniform Labeled Transition Systems (ULTraS) that capture different models of concurrent processes: fully nondeterministic (labeled transition systems, LTSs), fully probabilistic (labeled DTMCs), fully stochastic (labeled continuous time Markov chains, CTMCs), nondeterministic and probabilistic (Markov decision processes, MDPs), nondeterministic and stochastic (continuous time MDPs, CTMDPs). In [55], the bisimulation equivalences induced by some specific labeled state-to Function Transition Systems (FuTSs) were shown to coincide with the equivalences underlying the fragments of PEPA, Interactive Markov Language (IML) for Interactive Markov Chains (IMC) [44], Timed Process Calculus (TPC) [2] and Markov Automata Language (MAL) for Markov Automata Process Algebra (MAPA) [94]. In [61, 63], ordinary bisimulation (strong), quasi-lumping bisimulation (approximate strong) and proportional bisimulation equivalences on the PEPA components were investigated that induce, respectively, ordinary, quasi- and proportional lumpabilities on the corresponding CTMCs.

Nevertheless, no appropriate equivalence was defined for *parallel SPAs*. The non-interleaving bisimulation equivalence in Generalized Semi-Markovian Process Algebra (GSMPA) [28, 27] uses Start-Termination- (ST-) semantics for action particles while in Stochastic π-calculus (Sπ) [75] it is based on a sophisticated labeling.

1.5. **Our contributions.** As a basis model, we take *discrete time stochastic and deterministic Petri box calculus* (dtsdPBC), presented in [86, 87], featuring a step operational semantics. Here we do not consider the Petri net denotational semantics of the calculus, since it was extensively described in [86]. 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 [87], 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 [86], we proposed step stochastic bisimulation equivalence to identify algebraic processes with similar behaviour that are however differentiated by the semantics of the calculus. It enhances the corresponding relation from dtsiPBC, in that we now make difference between the states with positive sojourn times (tangible states) and those with zero sojourn times (vanishing states). Therefore, in the definition of the equivalence, we added a condition that vanishing states may only be related with vanishing states. We established consistency of the operational and denotational semantics of dtsdPBC up to step stochastic bisimulation equivalence. We examined the interrelations of the proposed notion with other equivalences of the algebra.

The main result of this paper is that step stochastic bisimulation equivalence can be used to reduce (by quotienting) the transition systems, SMCs, DTMCs and RDTMCs of the process expressions while preserving the qualitative and quantitative characteristics. We demonstrate isomorphism between the quotient transition systems of the process expressions and quotient reachability graphs of their dtsd-boxes. We also show that the quotient SMCs of the process expressions are isomorphic to those of their dtsd-boxes. We explore how the quotienting is related to extraction (of Markov chains from transition systems), embedding and reduction, by analyzing the transition probability matrices (TPMs) of the quotient DTMCs, EDTMCs and RDTMCs. In this way, we show that the reduced (by eliminating vanishing states) quotient TPMs coincide with the quotient reduced TPMs for

DTMCs of the process expressions. We prove that the mentioned equivalence guarantees identity of the stationary behaviour 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 behavioural structures. Hence, that *performance preserving equivalence* can help to reduce the number of states in the behaviour of a model and simplify its performance analysis that suffers from the state space explosion when large and complex realistic systems are modeled.

Thus, the main contributions of the paper are the following.

- Quotienting transition systems and Markov chains of the process expressions by step stochastic bisimulation equivalence to reduce the analysis complexity.
- Isomorphism of the quotient transition systems and reachability graphs, as well as the quotient SMCs, for the expressions and their dtsd-boxes.
- Permutability of the quotienting and reduction operations on DTMCs.
- Preservation of the stationary behaviour and residence time properties in the classes of step stochastic bisimulation equivalence of the expressions.
- Simplification of the performance evaluation in dtsdPBC by using the equivalence quotients of the transition systems, SMCs, DTMCs and RDTMCs.

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 behaviour and residence time properties in the equivalence classes. Section 7 summarizes the results obtained and outlines research perspectives in this area. The long and complex proofs are moved to Appendix A.

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 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 $\mathbb{N}^X_{fin}$. Let $M, M' \in \mathbb{N}^X_{fin}$. 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 $\mathcal{L} = \mathbb{N}^{\mathcal{A}}_{fin}$ be the set of *all multiactions*. Note that $\emptyset \in \mathcal{L}$ 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. We do not have probability 0 of stochastic multiactions, since they would not be performed in this case. 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 the corresponding operation in CCS [69].

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 in the usual way: 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 $f(\Upsilon) = \sum_{(\alpha,\kappa)\in\Upsilon}(f(\alpha), \kappa)$. The sums are considered with the multiplicity when applied to multisets: $f(\alpha) = \sum_{x\in\alpha} f(x) = \sum_{x\in\mathcal{A}} \alpha(x)f(x)$.

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

Let $\alpha, \beta \in \mathcal{L}$ be two multiactions such that for some elementary action $a \in Act$ we have $a \in \alpha$ and $\hat{a} \in \beta$, or $\hat{a} \in \alpha$ and $a \in \beta$. Then, synchronization of α and β by a is defined as $(\alpha \oplus_a \beta)(x) = \begin{cases} \alpha(x) + \beta(x) - 1, & x = a \text{ or } x = \hat{a}; \\ \alpha(x) + \beta(x), & \text{otherwise.} \end{cases}$
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 behaviour containing two concurrent activities that can be synchronized by a, there exists also the behaviour 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, and 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 [86]. A marking is the allocation of tokens in the places of a PN. Markings are used to describe dynamic behaviour 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. Therefore, 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 demonstrate irrelevance of the timer values for them (for example, their timers have not yet started or have already finished). The notion of the alphabet part for (the multisets of) stamped waiting multiactions is defined like that for (the multisets of) unstamped waiting multiactions.

By reasons of 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. *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*

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

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 [86] which are not safe, like shown for PNs in [22]. 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 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*

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

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 syntax) 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 $\delta \in \{1, \dots, \theta\}\}$. 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 behaviour. As for the Petri net intuition, dynamic expressions correspond to marked LDTSDPNs [86]. 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 $\delta \in \{1, \dots, \theta\}$ 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 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 [86] (but it is 2-bounded in the worst case, like shown for PNs in [22]).

Definition 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* $\lfloor G$ of G is obtained by removing from it 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 $\mathcal{SDL}(G) = \mathcal{SL}(G) \cup \mathcal{DL}(G) = \mathcal{SL}(G) \cup \mathcal{IL}(G) \cup \mathcal{WL}(G)$.

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 any discrete time tick or any transition firing in the corresponding LDTSDPN [86], 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 accomplished 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.

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

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

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*. By reason of brevity, two distinct inaction rules with the same premises are collated in some cases, resulting in the inaction rules with double conclusion.

Table 2. Inaction rules for arbitrary regular dynamic expressions

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

Definition 6. *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 7. *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 X be some set. We denote the Cartesian product $X \times X$ by X^2. Let $\mathcal{E} \subseteq X^2$ be an equivalence relation on X. Then the *equivalence class* (with respect to $\mathcal{E}$) of an element $x \in X$ is defined by $[x]_{\mathcal{E}} = \{y \in X \mid (x, y) \in \mathcal{E}\}$. The equivalence $\mathcal{E}$ partitions X into the *set of equivalence classes* $X/_{\mathcal{E}} = \{[x]_{\mathcal{E}} \mid x \in X\}$.

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 $H\}$.

Thus, 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 [86] 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 8. *An operative dynamic expression G is* saturated (with the values of timers)*, if each enabled waiting multiaction of $[G]_{\approx}$, being (certainly) 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 1. *Any operative dynamic expression can be always transformed into the saturated one by applying the inaction rules in a forward or a backward direction.*

Proof. See [86]. □

Thus, any dynamic expression can be always 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. Thus, 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 follows: $\overline{(\alpha, \natural_l^\theta)^\delta}$ is replaced with $\overline{(\alpha, \natural_l^\theta)^{\max\{1,\delta-1\}}}$ and $\underline{(\alpha, \natural_l^\theta)^\delta}$ is replaced with $\underline{(\alpha, \natural_l^\theta)^{\max\{1,\delta-1\}}}$ while $(\alpha, \natural_l^\theta)^\delta$ is replaced with $(\alpha, \natural_l^\theta)^{\max\{1,\delta-1\}}$.

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.

Observe that 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 $G \xrightarrow{\Gamma} \widetilde{G}$ ($G \xrightarrow{I} \widetilde{G}$ or $G \xrightarrow{W} \widetilde{G}$, 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 [86] 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 $G \xrightarrow{\emptyset} \circlearrowleft G$, 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 [86] 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 multiactions 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 9. *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. In the following, we suppose that the identical activities are enumerated when needed to avoid ambiguity. This enumeration is considered to be implicit.

Definition 10. *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) $\Upsilon - \{(\alpha,\kappa),(\beta,\lambda)\} + \{(\alpha \oplus_a \beta, \kappa\cdot\lambda)\} \in Can(G \text{ sy } a)$ *if* $\kappa,\lambda \in (0;1)$*;*

(b) $\Upsilon - \{(\alpha,\kappa),(\beta,\lambda)\} + \{(\alpha \oplus_a \beta, \natural_{l+m}^{\theta})\} \in Can(G \text{ sy } a)$ *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 executed 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 (besides that over stochastic 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 11. *Let* $G \in OpRegDynExpr$*. The* set of all non-empty multisets of activities which can be executed from G *is*

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

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 denote $\Upsilon_a=\{(\alpha,\kappa)\in\Upsilon\mid(a\in\alpha)\vee(\hat{a}\in\alpha)\}$.

We use the following abbreviations in the names of the rules from the table: "**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 in the table 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 with immediate multiactions have a suffix '**i**', and those with waiting multiactions have a suffix '**w**'. For explanation of the rules in Table 3, see [86].

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 [67, 1, 4, 5] 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.

Similar to [51], we are mainly 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 (i.e. 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 [86]. Formally, a dynamic expression G is *reachable*, if there exists a static expression E without timer value superscripts, such that $\overline{E}\approx G$ or $\overline{E}\approx G_0\stackrel{\Upsilon_1}{\to}H_1\approx G_1\stackrel{\Upsilon_2}{\to}\dots\stackrel{\Upsilon_n}{\to}H_n\approx G$ for some $\Upsilon_1,\dots,\Upsilon_n\in\mathbb{N}_{fin}^{\mathcal{SDL}}$.

Therefore, we consider a dynamic expression $G=\overline{(\{a\},\natural_1^2)^1[](\{b\},\natural_2^3)^1}$ as "illegal" and that $H=\overline{(\{a\},\natural_1^2)^1[](\{b\},\natural_2^3)^2}$ as "legal", since the latter is obtained from the overlined static expression without timer value superscripts $\overline{E}=\overline{(\{a\},\natural_1^2)[](\{b\},\natural_2^3)}$ after one time tick. On the other hand, G is "illegal" only when it is intended to specify a complete process, but it may become "legal" as a part of some complete specification, like $G\ \mathsf{rs}\ a$, since after two time ticks from $\overline{E\ \mathsf{rs}\ a}$, the timer values cannot be decreased further when the value 1 is approached. Thus, we should allow the dynamic expressions like G, by assuming that they are incomplete specifications, to be further composed. Further, a dynamic expression $G=\overline{(\{a\},\frac{1}{2})};(\{b\},\natural_1^2)^1$ is "illegal", since the waiting multiaction $(\{b\},\natural_1^2)$ is not enabled in $[G]_\approx$ and its timer cannot start before the stochastic multiaction $(\{a\},\frac{1}{2})$ is executed. Enabledness of the stamped waiting multiactions is considered in the next proposition.

Proposition 2. *Let G be a reachable dynamic expression. Then only waiting multiactions from $EnaWait([G]_\approx)$ are stamped in G.*

TABLE 3. Action and empty move rules

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

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

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

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

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

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

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

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

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

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

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

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

Proof. See [86]. □

3.3. **Transition systems.** We now construct labeled probabilistic transition systems associated with dynamic expressions. The transition systems are used to define the operational semantics of dynamic expressions.

Let G be a dynamic expression and $s = [G]_{\approx}$. The set of *all multisets of activities executable in* s is defined as $Exec(s) = \{\Upsilon \mid \exists H \in s\ \exists \widetilde{H}\ H \xrightarrow{\Upsilon} \widetilde{H}\}$. Here $H \xrightarrow{\Upsilon} \widetilde{H}$ 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 [86].

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 12. *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* $\exists \Upsilon\ H \stackrel{\Upsilon}{\rightarrow} \widetilde{H}$ *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 $DR(G) = DR_T(G) \uplus DR_V(G) = DR_{ST}(G) \uplus DR_{WT}(G) \uplus DR_V(G)$ ($\uplus$ 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

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

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

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

Note that the definition of $PF(\Upsilon, s)$ (and those of other probability functions we shall present) is based on the enumeration of activities which is considered implicit.

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

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

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

Note that $\forall s \in DR(G)\ \sum_{\{\tilde{s} \mid \exists H \in s\ \exists \widetilde{H} \in \tilde{s}\ \exists \Upsilon\ H \stackrel{\Upsilon}{\rightarrow} \widetilde{H}\}} PM(s, \tilde{s}) =$
$\sum_{\{\tilde{s} \mid \exists H \in s\ \exists \widetilde{H} \in \tilde{s}\ \exists \Upsilon\ H \stackrel{\Upsilon}{\rightarrow} \widetilde{H}\}} \sum_{\{\Upsilon \mid \exists H \in s\ \exists \widetilde{H} \in \tilde{s}\ H \stackrel{\Upsilon}{\rightarrow} \widetilde{H}\}} PT(\Upsilon, s) = \sum_{\Upsilon \in Exec(s)} PT(\Upsilon, s) = 1.$

Definition 13. *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 definition of $TS(G)$ is correct, i.e. for every state, the sum of the probabilities of all the transitions starting from it is 1, by the note after the definition of $PT(\Upsilon, s)$.

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 in the case when we cannot leave an s-tangible state s and the only transition leaving it is the empty move one $s \xrightarrow{\emptyset}_1 \circlearrowleft s$, or if there is a single transition from a w-tangible or a vanishing state to any other one. 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 14. *Let G, G' be dynamic expressions and $TS(G) = (S_G, L_G, \mathcal{T}_G, s_G)$, $TS(G') = (S_{G'}, L_{G'}, \mathcal{T}_{G'}, s_{G'})$ be their transition systems. A mapping $\beta : S_G \rightarrow S_{G'}$ is an* isomorphism *between $TS(G)$ and $TS(G')$, denoted by $\beta : TS(G) \simeq TS(G')$, if*

(1) *β is a bijection such that $\beta(s_G) = s_{G'}$;*

(2) $\forall s, \tilde{s} \in S_G\ \forall \Upsilon\ s \xrightarrow{\Upsilon}_{\mathcal{P}} \tilde{s} \ \Leftrightarrow\ \beta(s) \xrightarrow{\Upsilon}_{\mathcal{P}} \beta(\tilde{s})$.

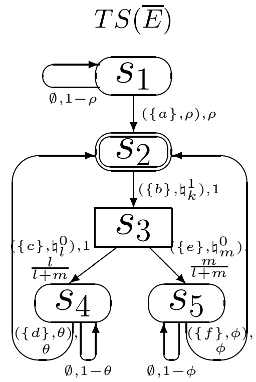

FIG. 1. The transition system of $\overline{E}$ for $E = [(\{a\},\rho) * ((\{b\},\natural_k^1); (((\{c\},\natural_l^0);(\{d\},\theta))[]((\{e\},\natural_m^0);(\{f\},\phi)))) * \mathsf{Stop}]$

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

Example 1. *The expression* $\mathsf{Stop} = (\{g\},\frac{1}{2})$ rs g *specifies the non-terminating process that performs only empty loops with probability* 1.

Let $E = [(\{a\},\rho) * ((\{b\},\natural_k^1); (((\{c\},\natural_l^0);(\{d\},\theta))[]((\{e\},\natural_m^0);(\{f\},\phi)))) * \mathsf{Stop}]$, *where* $\rho,\theta,\phi \in (0;1)$ *and* $k,l,m \in \mathbb{R}_{>0}$. $DR(\overline{E})$ *consists of the elements*

$$
\begin{aligned}
s_1 &= [[\overline{(\{a\},\rho)} * ((\{b\},\natural_k^1); (((\{c\},\natural_l^0);(\{d\},\theta))[]((\{e\},\natural_m^0);(\{f\},\phi)))) * \mathsf{Stop}]]_\approx,\\
s_2 &= [[(\{a\},\rho) * \overline{((\{b\},\natural_k^1)^1}; (((\{c\},\natural_l^0);(\{d\},\theta))[]((\{e\},\natural_m^0);(\{f\},\phi)))) * \mathsf{Stop}]]_\approx,\\
s_3 &= [[(\{a\},\rho) * ((\{b\},\natural_k^1); ((\overline{(\{c\},\natural_l^0)};(\{d\},\theta))[]((\{e\},\natural_m^0);(\{f\},\phi)))) * \mathsf{Stop}]]_\approx =\\
&\ [[(\{a\},\rho) * ((\{b\},\natural_k^1); (((\{c\},\natural_l^0);(\{d\},\theta))[](\overline{(\{e\},\natural_m^0)};(\{f\},\phi)))) * \mathsf{Stop}]]_\approx,\\
s_4 &= [[(\{a\},\rho) * ((\{b\},\natural_k^1); (((\{c\},\natural_l^0);\overline{(\{d\},\theta))}[]((\{e\},\natural_m^0);(\{f\},\phi)))) * \mathsf{Stop}]]_\approx,\\
s_5 &= [[(\{a\},\rho) * ((\{b\},\natural_k^1); (((\{c\},\natural_l^0);(\{d\},\theta))[]((\{e\},\natural_m^0);\overline{(\{f\},\phi)))})) * \mathsf{Stop}]]_\approx.
\end{aligned}
$$

We have $DR_{ST}(\overline{E}) = \{s_1, s_4, s_5\}$, $DR_{WT}(\overline{E}) = \{s_2\}$ *and* $DR_V(\overline{E}) = \{s_3\}$.

In Figure 1, the transition system $TS(\overline{E})$ is presented. The s-tangible and w-tangible states are depicted in ordinary and double ovals, respectively, and the vanishing ones are depicted in boxes.

Example 2. *Let us interpret E from Example 1 as a specification of the travel system. A tourist visits regularly new cities. After seeing the sights of the current city, he goes to the next city by the nearest train or bus available at the city station. Buses depart less frequently than trains, but the next city is quicker reached by bus than by train. We suppose that the stay duration in every city (being a constant), the departure numbers of trains and buses, as well as their speeds* do not depend *on a particular city, bus or train. The travel route has been planned so that the distances between successive cities* coincide.

The meaning of actions and activities from the syntax of E is as follows. The action a corresponds to the system activation after planning the travel route that takes *a time, geometrically distributed with a parameter ρ, the probability of the corresponding stochastic multiaction $(\{a\},\rho)$. The action b represents coming to the city station after completion of looking round the current city that takes (for every city) a fixed time equal to 1 (hour), the time delay of the corresponding waiting multiaction $(\{b\},\natural_k^1)$ with (resolving no choice) weight k. The actions c and e correspond to the urgent (in zero time) getting on bus and train, respectively, and thus model the choice between these two transport facilities. The weights of the two corresponding immediate multiactions $(\{c\},\natural_l^0)$ and $(\{e\},\natural_m^0)$ suggest that every l departures of buses take the same time as m departures of trains ($l<m$), hence, a bus departs with the probability $\frac{l}{l+m}$ while a train departs with the probability $\frac{m}{l+m}$. The actions d and f correspond to coming in a city by bus and train, respectively, that takes a time, geometrically distributed with the parameters θ and ϕ, respectively ($\theta>\phi$), the probabilities of the corresponding stochastic multiactions $(\{d\},\theta)$ and $(\{f\},\phi)$.*

The meaning of states from $DR(\overline{E})$ is the following. The s-tangible state s_1 corresponds to staying at home and planning the future travel. The w-tangible state s_2 means residence in a city for exactly one time unit (hour). The vanishing state s_3 with zero residence time represents instantaneous stay at the city station, signifying that the tourist does not wait there for departure of the transport. The s-tangible states s_4 and s_5 correspond to going by bus and train, respectively.

In Example 3 from [87]*, we calculated the following performance indices, based on the steady-state probability mass function (PMF) for the underlying SMC of $\overline{E}$ $SMC(\overline{E})$ $\varphi=\frac{1}{\theta\phi(l+m)+\phi l+\theta m}(0,\theta\phi(l+m),0,\phi l,\theta m)$ and the average sojourn time vector of $\overline{E}$ $SJ=\left(\frac{1}{\rho},1,0,\frac{1}{\theta},\frac{1}{\phi}\right)$.*

- *The average time between comings to the successive cities (mean sightseeing and travel time) is $ReturnTime(s_2)=\frac{1}{\varphi(s_2)}=1+\frac{\phi l+\theta m}{\theta\phi(l+m)}$.*
- *The fraction of time spent in a city (sightseeing time fraction) is $TimeFract(s_2)=\varphi(s_2)=\frac{\theta\phi(l+m)}{\theta\phi(l+m)+\phi l+\theta m}$.*
- *The fraction of time spent in a transport (travel time fraction) is $TimeFract(\{s_4,s_5\})=\varphi(s_4)+\varphi(s_5)=\frac{\phi l+\theta m}{\theta\phi(l+m)+\phi l+\theta m}$.*
- *The relative fraction of time spent in a city with respect to that spent in transport (sightseeing relative to travel time fraction) is $RltTimeFract(\{s_2\},\{s_4,s_5\})=\frac{\varphi(s_2)}{\varphi(s_4)+\varphi(s_5)}=\frac{\theta\phi(l+m)}{\phi l+\theta m}$.*
- *The rate of leaving/entering a city (departure/arrival rate) is $ExitFreq(s_2)=\frac{\varphi(s_2)}{SJ(s_2)}=\frac{\theta\phi(l+m)}{\theta\phi(l+m)+\phi l+\theta m}$.*

Let $N=(P_N,T_N,W_N,D_N,\Omega_N,\mathcal{L}_N,Q_N)$ be a LDTSDPN [86] 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 corresponding notions for dynamic expressions in [87]. Every marked and clocked plain dtsd-box [86] can be interpreted as an LDTSDPN. Therefore, we can evaluate performance with the LDTSDPNs corresponding to dtsd-boxes and then transfer the results to the latter.

Example 3. *Let E be from Example 1 and N be the marked and clocked dtsd-box of $\overline{E}$, denoted by $N=Box_{dtsd}(\overline{E})$* [86]*. In Figure 2, the underlying SMC $SMC(N)$*

$SMC(N)$

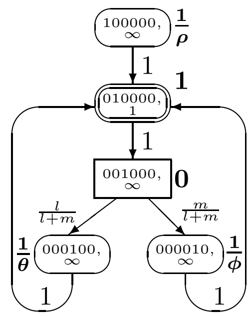

FIG. 2. The underlying SMC of $N = Box_{dtsd}(\overline{E})$ for $E = [(\{a\}, \rho) * ((\{b\}, \natural_k^1); (((\{c\}, \natural_l^0); (\{d\}, \theta))[](((\{e\}, \natural_m^0); (\{f\}, \phi)))) * \mathsf{Stop}]$

is presented. Note that $SMC(\overline{E})$ [87] and $SMC(N)$ are isomorphic. Thus, both the transient and steady-state PMFs for $SMC(N)$ and $SMC(\overline{E})$ coincide.

4. Stochastic equivalences

The semantic equivalence $=_{ts}$ is too discriminating in many cases, hence, we need weaker equivalence notions. These equivalences should possess the following necessary properties. First, any two equivalent processes must have the same sequences of multisets of multiactions, which are the multiaction parts of the activities executed in steps starting from the initial states of the processes. Second, for every such sequence, its execution probabilities within both processes must coincide. Third, the desired equivalence should preserve the branching structure of computations, i.e. the points of choice of an external observer between several extensions of a particular computation should be taken into account. In this section, we define 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 [50, 54, 45, 46, 18, 10, 11], 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 [50], in contrast to the reactive model, treated in [54], and we take transition probabilities instead of transition rates from [45, 46, 18, 10, 11]. Thus, step stochastic bisimulation equivalence that we define further is (in the probabilistic sense) comparable only with interleaving probabilistic bisimulation equivalence from [50], and our relation is obviously stronger.

In the definition below, we consider $\mathcal{L}(\Upsilon) \in \mathbb{N}_{fin}^{\mathcal{L}}$ for $\Upsilon \in \mathbb{N}_{fin}^{\mathcal{SIL}}$, 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 $\mathcal{H} \subseteq DR(G)$. Then, for any $s \in DR(G)$ and $A \in \mathbb{N}_{fin}^{\mathcal{L}}$, we write $s \xrightarrow{A}_{\mathcal{P}} \mathcal{H}$, where $\mathcal{P} = PM_A(s, \mathcal{H})$ is the *overall probability to move from s into the set of states $\mathcal{H}$ via steps with the multiaction part A* defined as

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

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 $\exists \mathcal{P}\ s \xrightarrow{A}_{\mathcal{P}} \tilde{s}$.

Definition 16. *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*
$\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}\ \forall A \in \mathbb{N}_{fin}^{\mathcal{L}}\ s_1 \xrightarrow{A}_{\mathcal{P}} \mathcal{H} \ \Leftrightarrow\ s_2 \xrightarrow{A}_{\mathcal{P}} \mathcal{H}$.

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 17. *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*

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

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

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

$TS_{\mathcal{L}}(\overline{F})$

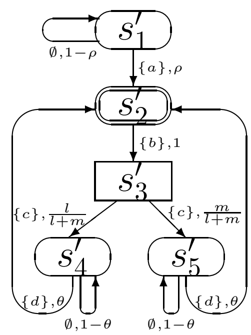

FIG. 3. The multiaction transition system of $\overline{F}$ for $F = [(\{a\},\rho) * ((\{b\},\natural_k^1);(((\{c\},\natural_l^0);(\{d\},\theta)_1)[](((\{c\},\natural_m^0);(\{d\},\theta)_2))) * \mathsf{Stop}]$

analogy (excepting step semantics) with interleaving probabilistic bisimulation on generative probabilistic transition systems from [50].

Example 4. *Let us consider an abstraction F of the static expression E from Example 1, such that $c = e,\ d = f,\ \theta = \phi$, i.e. $F = [(\{a\},\rho) * ((\{b\},\natural_k^1);(((\{c\},\natural_l^0);(\{d\},\theta)_1)[](((\{c\},\natural_m^0);(\{d\},\theta)_2))) * \mathsf{Stop}]$. Then $DR(\overline{F}) = \{s_1', s_2', s_3', s_4', s_5'\}$ is obtained from $DR(\overline{E})$ via substitution of the symbols $e,\ f,\ \phi$ by $c,\ d,\ \theta$, respectively, in the specifications of the corresponding states from the latter set. We have $DR_{ST}(\overline{F})= \{s_1', s_4', s_5'\},\ DR_{WT}(\overline{F}) = \{s_2'\}$ and $DR_V(\overline{F}) = \{s_3'\}$. In Figure 3, the multiaction transition system $TS_{\mathcal{L}}(\overline{F})$ is presented. To simplify the presentation, the singleton multisets of multiactions are written without outer braces.*

Example 5. *Let us interpret F from Example 4 as an abstraction of the travel system from Example 2. In such an* abstract *travel system, we do not differentiate between the transport facilities (trains or buses) that always have the same speed, but every l departures of the transport from the first platform take the same time as m departures of the transport from the second platform, and the traveler can choose between the two platforms.*

By taking $\theta = \phi$ in Example 2, we now calculate the following performance indices, based on the steady-state PMF for $SMC(\overline{F})$ $\varphi = \frac{1}{1+\theta}\left(0, \theta, 0, \frac{l}{l+m}, \frac{m}{l+m}\right)$ and the average sojourn time vector of $\overline{F}$ $SJ = \left(\frac{1}{\rho}, 1, 0, \frac{1}{\theta}, \frac{1}{\theta}\right)$.

- *The average time between comings to the successive cities (mean sightseeing and travel time) is $ReturnTime(s_2') = \frac{1}{\varphi(s_2')} = 1 + \frac{\theta l + \theta m}{\theta^2(l+m)} = 1 + \frac{1}{\theta}$.*
- *The fraction of time spent in a city (sightseeing time fraction) is $TimeFract(s_2') = \varphi(s_2') = \frac{\theta^2(l+m)}{\theta^2(l+m)+\theta l+\theta m} = \frac{\theta}{1+\theta}$.*
- *The fraction of time spent in a transport (travel time fraction) is $TimeFract(\{s_4', s_5'\}) = \varphi(s_4') + \varphi(s_5') = \frac{\theta l+\theta m}{\theta^2(l+m)+\theta l+\theta m} = \frac{1}{1+\theta}$.*
- *The relative fraction of time spent in a city with respect to that spent in transport (sightseeing relative to travel time fraction) is $RltTimeFract(\{s_2'\}, \{s_4', s_5'\}) = \frac{\varphi(s_2')}{\varphi(s_4')+\varphi(s_5')} = \frac{\theta^2(l+m)}{\theta l+\theta m} = \theta$.*

- *The rate of leaving/entering a city (departure/arrival rate) is* $ExitFreq(s_2') = \frac{\varphi(s_2')}{SJ(s_2')} = \frac{\theta^2(l+m)}{\theta^2(l+m)+\theta l+\theta m} = \frac{\theta}{1+\theta}$.

The following proposition states that every step stochastic bisimulation binds s-tangible states only with s-tangible ones, and the same is valid for w-tangible states, as well as for vanishing states.

Proposition 3. *Let G and G' be dynamic expressions and $\mathcal{R} : G\underline{\leftrightarrow}_{ss}G'$. Then* $\mathcal{R}\subseteq(DR_{ST}(G)\cup DR_{ST}(G'))^2\uplus(DR_{WT}(G)\cup DR_{WT}(G'))^2\uplus(DR_V(G)\cup DR_V(G'))^2$.

Proof. See [86]. □

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

Proof. See [86]. □

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

Theorem 1. *For any static expression E, $TS(\overline{E})\underline{\leftrightarrow}_{ss}RG(Box_{dtsd}(\overline{E}))$.*

Proof. See [86]. □

We now compare the discrimination power of the stochastic equivalences.

Theorem 2. *For dynamic expressions G and G' the next* strict *implications hold:*

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

Proof. See [86]. □

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 behavioural properties. Thus, the reduction allows one to simplify the functional and performance analysis.

5.1. Quotients of the transition systems and Markov chains.

We now consider the quotient transition systems and Markov chains (SMCs, DTMCs, RDTMCs).

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

By the note after Proposition 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, $\mathcal{R}_{ss}(G)$ 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 the quotient (by $\underline{\leftrightarrow}_{ss}$) underlying SMCs of expressions can be defined. The mentioned 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 18. *Let G be a dynamic expression. The* quotient (by $\underline{\leftrightarrow}_{ss}$) (labeled probabilistic) transition system *of G is a quadruple $TS_{\underline{\leftrightarrow}_{ss}}(G) = (S_{\underline{\leftrightarrow}_{ss}}, L_{\underline{\leftrightarrow}_{ss}}, \mathcal{T}_{\underline{\leftrightarrow}_{ss}}, s_{\underline{\leftrightarrow}_{ss}})$, where*

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

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

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 $\mathcal{L}_i = \mathcal{K}_i \cup \{\mathcal{K}_i\}$ $(1 \le i \le n)$, 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 [50]. 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 $TS_{\underline{\leftrightarrow}_{ss}}(G)$.

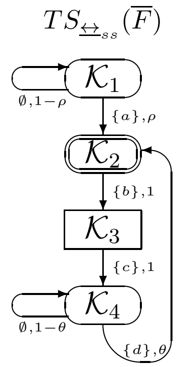

FIG. 4. The quotient transition system of $\overline{F}$ for $F = [(\{a\},\rho) * ((\{b\},\natural_k^1);((((\{c\},\natural_l^0);(\{d\},\theta)_1)[]((\{c\},\natural_m^0);(\{d\},\theta)_2))) * \mathsf{Stop}]$

Example 6. *Let F be from Example 4. Then $DR(\overline{F})/_{\mathcal{R}_{ss}(\overline{F})} = \{\mathcal{K}_1,\mathcal{K}_2,\mathcal{K}_3,\mathcal{K}_4\}$, where $\mathcal{K}_1=\{s'_1\}$, $\mathcal{K}_2=\{s'_2\}$, $\mathcal{K}_3=\{s'_3\}$, $\mathcal{K}_4=\{s'_4,s'_5\}$. We have $DR_{ST}(\overline{F})/_{\mathcal{R}_{ss}(\overline{F})} = \{\mathcal{K}_1,\mathcal{K}_4\}$, $DR_{WT}(\overline{F})/_{\mathcal{R}_{ss}(\overline{F})} = \{\mathcal{K}_2\}$ and $DR_V(\overline{F})/_{\mathcal{R}_{ss}(\overline{F})} = \{\mathcal{K}_3\}$. Thus, $\mathcal{R}_{ss}$ merges the states with the same "futures" from the different branches. In Figure 4, the quotient transition system $TS_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is presented.*

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 establishes a connection between quotient (by $\underline{\leftrightarrow}_{ss}$) transition systems of the overlined static expressions and quotient reachability graphs of their dtsd-boxes.

Proposition 5. *For any static expression E,*

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

Proof. By definitions of the quotient (by $\underline{\leftrightarrow}_{ss}$) transition systems and quotient reachability graphs, their uniqueness up to isomorphism and Theorem 1. □

Example 7. *Let F be from Example 4 and $N' = Box_{dtsd}(\overline{F})$. Then $RS(N')/_{\mathcal{R}_{ss}(N')} = \{\mathcal{L}_1,\mathcal{L}_2,\mathcal{L}_3,\mathcal{L}_4\}$, where $\mathcal{L}_1 = \{Q'_1\}$, $\mathcal{L}_2 = \{Q'_2\}$, $\mathcal{L}_3 = \{Q'_3\}$, $\mathcal{L}_4 = \{Q'_4,Q'_5\}$ for $Q'_1 = ((1,0,0,0,0,0),\infty)$, $Q'_2 = ((0,1,0,0,0,0),1)$, $Q'_3 = ((0,0,1,0,0,0),\infty)$, $Q'_4 = ((0,0,0,1,0,0),\infty)$, $Q'_5 = ((0,0,0,0,1,0),\infty)$. We have $RS_{ST}(N')/_{\mathcal{R}_{ss}(N')}=\{\mathcal{L}_1,\mathcal{L}_4\}$, $RS_{WT}(N')/_{\mathcal{R}_{ss}(N')}=\{\mathcal{L}_2\}$ and $RS_V(N')/_{\mathcal{R}_{ss}(N')}=\{\mathcal{L}_3\}$. In Figure 5, the quotient reachability graph $RG_{\underline{\leftrightarrow}_{ss}}(N')$ is presented. Note that $TS_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ and $RG_{\underline{\leftrightarrow}_{ss}}(N')$ are isomorphic.*

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

$$PM(\mathcal{K},\mathcal{K})^k.$$

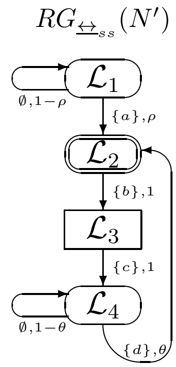

FIG. 5. The quotient reachability graph of $N' = Box_{dtsd}(\overline{F})$ for $F = [(\{a\},\rho) * ((\{b\},\natural_k^1);(((\{c\},\natural_l^0);(\{d\},\theta)_1)[]((\{c\},\natural_m^0);(\{d\},\theta)_2))) * \mathsf{Stop}]$

The *quotient (by $\underline{\leftrightarrow}_{ss}$) self-loops abstraction factor in the equivalence class $\mathcal{K}$* is

$$SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) = \begin{cases} \frac{1}{1-PM(\mathcal{K},\mathcal{K})}, & \mathcal{K} \to \mathcal{K}; \\ 1, & \text{otherwise.} \end{cases}$$

The *quotient (by $\underline{\leftrightarrow}_{ss}$) self-loops abstraction vector* of G, denoted by $SL_{\underline{\leftrightarrow}_{ss}}$, has the elements $SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}),\ \mathcal{K} \in DR(G)/_{\mathcal{R}_{ss}(G)}$.

Let $\mathcal{K} \to \widetilde{\mathcal{K}}$ 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

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

The value $k=0$ in the summation 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 $\mathcal{K} \in DR_V(G)/_{\mathcal{R}_{ss}(G)}$. We have $\forall \mathcal{K} \in DR_V(G)/_{\mathcal{R}_{ss}(G)}\ SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) \neq SJ_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) = 0$ 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 19. *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.*

The steady-state probability mass functions (PMFs) $\psi^*_{\underline{\leftrightarrow}_{ss}}$ for $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ and $\varphi_{\underline{\leftrightarrow}_{ss}}$ for $SMC_{\underline{\leftrightarrow}_{ss}}(G)$ are defined like the respective notions ψ^* for $EDTMC(G)$ and φ for $SMC(G)$ [87].

Example 8. *Let F be from Example 6. In Figure 6, the quotient underlying SMC $SMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is presented. The average sojourn times in the states of the underlying quotient SMC are written next to them in bold font.*

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

$$SJ_{\underline{\leftrightarrow}_{ss}} = \left(\frac{1}{\rho}, 1, 0, \frac{1}{\theta}\right).$$

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

$$VAR_{\underline{\leftrightarrow}_{ss}} = \left(\frac{1-\rho}{\rho^2}, 0, 0, \frac{1-\theta}{\theta^2}\right).$$

The transition probability matrix (TPM) for $EDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is

$$\mathbf{P}^*_{\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

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

$$\psi^*_{\underline{\leftrightarrow}_{ss}} = \left(0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right).$$

The steady-state PMF $\psi^*_{\underline{\leftrightarrow}_{ss}}$ weighted by $SJ_{\underline{\leftrightarrow}_{ss}}$ is

$$\left(0, \frac{1}{3}, 0, \frac{l}{3\theta}\right).$$

It remains to normalize the steady-state weighted PMF by dividing it by the sum of its components

$$\psi^*_{\underline{\leftrightarrow}_{ss}} SJ^T_{\underline{\leftrightarrow}_{ss}} = \frac{1+\theta}{3\theta}.$$

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

$$\varphi_{\underline{\leftrightarrow}_{ss}} = \frac{1}{1+\theta}(0, \theta, 0, 1).$$

$SMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$

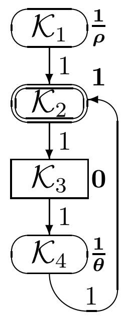

FIG. 6. The quotient underlying SMC of $\overline{F}$ for $F = [(\{a\}, \rho) * ((\{b\}, \natural_k^1); (((\{c\}, \natural_l^0); (\{d\}, \theta)_1)[]((\{c\}, \natural_m^0); (\{d\}, \theta)_2))) * \mathsf{Stop}]$

Example 9. *Let F be from Example 4. We now calculate the following performance indices, based on the steady-state PMF for $SMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ $\varphi_{\underline{\leftrightarrow}_{ss}} = \frac{1}{1+\theta}(0, \theta, 0, 1)$ and the quotient average sojourn time vector of $\overline{F}$ $SJ_{\underline{\leftrightarrow}_{ss}} = \left(\frac{1}{\rho}, 1, 0, \frac{1}{\theta}\right)$.*

- *The average time between comings to the successive cities (mean sightseeing and travel time) is $ReturnTime(\mathcal{K}_2) = \frac{1}{\varphi(\mathcal{K}_2)} = 1 + \frac{1}{\theta}$.*
- *The fraction of time spent in a city (sightseeing time fraction) is $TimeFract(\mathcal{K}_2) = \varphi(\mathcal{K}_2) = \frac{\theta}{1+\theta}$.*
- *The fraction of time spent in a transport (travel time fraction) is $TimeFract(\mathcal{K}_4) = \varphi(\mathcal{K}_4) = \frac{1}{1+\theta}$.*
- *The relative fraction of time spent in a city with respect to that spent in transport (sightseeing relative to travel time fraction) is $RltTimeFract(\{\mathcal{K}_2\}, \{\mathcal{K}_4\}) = \frac{\varphi(\mathcal{K}_2)}{\varphi(\mathcal{K}_4)} = \theta$.*
- *The rate of leaving/entering a city (departure/arrival rate) is $ExitFreq(\mathcal{K}_2) = \frac{\varphi(\mathcal{K}_2)}{SJ(\mathcal{K}_2)} = \frac{\theta}{1+\theta}$.*

The performance indices are the same for the "complete" and the "quotient" abstract travel systems. The coincidence of the indices will illustrate the results of the forthcoming Proposition 10 and Proposition 11 (both modified for $\mathcal{R}_{\mathcal{L}ss}(\overline{F})$).

Let $\simeq$ denote isomorphism between SMCs that binds their initial states, where two SMCs are isomorphic if their EDTMCs are so and the sojourn times in the isomorphic states of the EDTMCs 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 6. *For any static expression E*

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

Proof. By definitions of the quotient (by $\underline{\leftrightarrow}_{ss}$) underlying SMCs for dynamic expressions and LDTSDPNs and Proposition 5, taking into account the following. First, for the associated SMCs, the average sojourn time in the states is the same, since it is defined via the analogous probability functions. Second, the transition

$SMC_{\underline{\leftrightarrow}_{ss}}(N')$

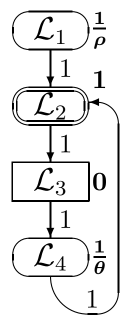

FIG. 7. The quotient underlying SMC of $N' = Box_{dtsd}(\overline{F})$ for $F = [(\{a\},\rho) * ((\{b\},\natural_k^1);(((\{c\},\natural_l^0);(\{d\},\theta)_1)[]((\{c\},\natural_m^0);(\{d\},\theta)_2))) * \mathsf{Stop}]$

probabilities of the associated SMCs are the sums of those belonging to the quotient transition systems or the quotient reachability graphs.

For instance, observe that the probability functions $PM(\mathcal{K},\widetilde{\mathcal{K}})$ and $PM^*(\mathcal{K},\widetilde{\mathcal{K}})$ can be respectively defined in the same way as $PM(\mathcal{L},\widetilde{\mathcal{L}})$ and $PM^*(\mathcal{L},\widetilde{\mathcal{L}})$, for the corresponding equivalence classes of the process states and net states $\mathcal{K}$ and $\mathcal{L}$, as well as $\widetilde{\mathcal{K}}$ and $\widetilde{\mathcal{L}}$. $\square$

Example 10. *Let F be from Example 4 and $N' = Box_{dtsd}(\overline{F})$. In Figure 7, the quotient underlying SMC $SMC_{\underline{\leftrightarrow}_{ss}}(N')$ is presented. Note that $SMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ and $SMC_{\underline{\leftrightarrow}_{ss}}(N')$ are isomorphic. Thus, both the transient and steady-state PMFs for $SMC_{\underline{\leftrightarrow}_{ss}}(N')$ and $SMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ coincide.*

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. Such reduction method resembles that from [3] based on place bisimulation equivalence for PNs, excepting that the former method merges states, while the latter one merges places.

Moreover, the algorithms exist to construct the quotients of transition systems by an equivalence (like bisimulation one) [74] and those of (discrete or continuous time) Markov chains by ordinary lumping [38]. The algorithms have time complexity $O(m \log n)$ and space complexity $O(m+n)$, where n is the number of states and m is the number of transitions. As mentioned in [97], the algorithm from [38] can be easily adjusted to produce quotients of labeled probabilistic transition systems by the probabilistic bisimulation equivalence. In [97], the symbolic partition refinement algorithm on state space of CTMCs was proposed. The algorithm can be straightforwardly accommodated to DTMCs, interactive Markov chains (IMCs), Markov reward models, Markov decision processes (MDPs), Kripke structures and labeled probabilistic transition systems. Such a symbolic lumping uses memory efficiently due to compact representation of the state space partition. The symbolic lumping is time efficient, since fast algorithm of the partition representation and refinement is applied. In [39], a polynomial-time algorithm for minimizing behaviour of probabilistic automata by probabilistic bisimulation equivalence was outlined that results in the canonical quotient structures. One can adapt the above algorithms for our framework of transition systems, (reduced) DTMCs and SMCs.

Let us consider quotient (by $\underline{\leftrightarrow}_{ss}$) DTMCs of expressions based on the state change probabilities $PM(\mathcal{K}, \widetilde{\mathcal{K}})$.

Definition 20. *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 corresponding notion ψ for $DTMC(G)$ [87].

Example 11. *Let F be from Example 6. In Figure 8, the quotient DTMC $DTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is presented.*

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

$$\mathbf{P}_{\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} 1-\rho & \rho & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \theta & 0 & 1-\theta \end{pmatrix}.$$

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

$$\psi_{\underline{\leftrightarrow}_{ss}} = \frac{1}{1+2\theta}(0, \theta, \theta, 1).$$

Remember that $DR_T(\overline{F})/_{\mathcal{R}_{ss}(F)} = DR_{ST}(\overline{F})/_{\mathcal{R}_{ss}(F)} \cup DR_{WT}(\overline{F})/_{\mathcal{R}_{ss}(F)} = \{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_4\}$ and $DR_V(\overline{F})/_{\mathcal{R}_{ss}(F)} = \{\mathcal{K}_3\}$. Hence,

$$\sum_{\mathcal{K} \in DR_T(\overline{F})/_{\mathcal{R}_{ss}(F)}} \psi(\mathcal{K}) = \psi(\mathcal{K}_1) + \psi(\mathcal{K}_2) + \psi(\mathcal{K}_4) = \frac{1+\theta}{1+2\theta}.$$

By the "quotient" analogue of Proposition 4 from [87], *we have*

$$\begin{aligned} &\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_1) = 0 \cdot \tfrac{1+2\theta}{1+\theta} = 0, \\ &\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_2) = \tfrac{\theta}{1+2\theta} \cdot \tfrac{1+2\theta}{1+\theta} = \tfrac{\theta}{1+\theta}, \\ &\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_3) = 0, \\ &\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_4) = \tfrac{1}{1+2\theta} \cdot \tfrac{1+2\theta}{1+\theta} = \tfrac{1}{1+\theta}. \end{aligned}$$

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

$$\varphi_{\underline{\leftrightarrow}_{ss}} = \frac{1}{1+\theta}(0, \theta, 0, 1).$$

This coincides with the result obtained in Example 8 with the use of $\psi^*_{\underline{\leftrightarrow}_{ss}}$ and $SJ_{\underline{\leftrightarrow}_{ss}}$.

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 21. *The* reduced quotient (by $\underline{\leftrightarrow}_{ss}$) DTMC *of G, denoted by $RDTMC_{\underline{\leftrightarrow}_{ss}}(G)$, is defined like $RDTMC(G)$ in* [87], *but it is constructed from $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$ instead of $DTMC(G)$.*

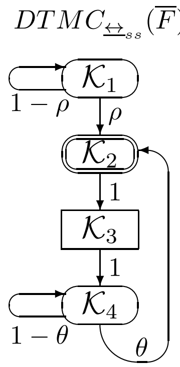

FIG. 8. The quotient DTMC of $\overline{F}$ for $F = [(\{a\}, \rho) * ((\{b\}, \natural_k^1); ((((\{c\}, \natural_l^0); (\{d\}, \theta)_1))[]((\{c\}, \natural_m^0); (\{d\}, \theta)_2))) * \mathsf{Stop}]$

The steady-state PMF $\psi^\diamond_{\underline{\leftrightarrow}_{ss}}$ for $RDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ is defined like the corresponding notion $\psi^\diamond$ for $RDTMC(G)$ [87].

Example 12. *Let F be from Example 6. Remember that $DR_T(\overline{F})/_{\mathcal{R}_{ss}(F)} = DR_{ST}(\overline{F})/_{\mathcal{R}_{ss}(F)} \cup DR_{WT}(\overline{F})/_{\mathcal{R}_{ss}(F)} = \{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_4\}$ and $DR_V(\overline{F})/_{\mathcal{R}_{ss}(F)} = \{\mathcal{K}_3\}$. We reorder the states from $DR(\overline{F})/_{\mathcal{R}_{ss}(F)}$, by moving vanishing states to the first positions: $\mathcal{K}_3, \mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_4$.*

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

$$\mathbf{P}_{r\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1-\rho & \rho & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & \theta & 1-\theta \end{pmatrix}.$$

The result of the decomposing $\mathbf{P}_{r\underline{\leftrightarrow}_{ss}}$ are the matrices

$$\mathbf{C}_{\underline{\leftrightarrow}_{ss}} = 0,\ \mathbf{D}_{\underline{\leftrightarrow}_{ss}} = (0,0,1),\ \mathbf{E}_{\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},\ \mathbf{F}_{\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} 1-\rho & \rho & 0 \\ 0 & 0 & 0 \\ 0 & \theta & 1-\theta \end{pmatrix}.$$

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

$$\mathbf{G}_{\underline{\leftrightarrow}_{ss}} = \sum_{k=0}^{l} \mathbf{C}^k_{\underline{\leftrightarrow}_{ss}} = \mathbf{C}^0_{\underline{\leftrightarrow}_{ss}} = \mathbf{I}.$$

Further, the TPM for $RDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is $\mathbf{P}^\diamond_{\underline{\leftrightarrow}_{ss}} = \mathbf{F}_{\underline{\leftrightarrow}_{ss}} + \mathbf{E}_{\underline{\leftrightarrow}_{ss}}\mathbf{G}_{\underline{\leftrightarrow}_{ss}}\mathbf{D}_{\underline{\leftrightarrow}_{ss}} =$

$$\mathbf{F}_{\underline{\leftrightarrow}_{ss}} + \mathbf{E}_{\underline{\leftrightarrow}_{ss}}\mathbf{I}\mathbf{D}_{\underline{\leftrightarrow}_{ss}} = \mathbf{F}_{\underline{\leftrightarrow}_{ss}} + \mathbf{E}_{\underline{\leftrightarrow}_{ss}}\mathbf{D}_{\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} 1-\rho & \rho & 0 \\ 0 & 0 & 1 \\ 0 & \theta & 1-\theta \end{pmatrix}.$$

In Figure 9, the reduced quotient DTMC $RDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is presented. The steady-state PMF for $RDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is

$$\psi^\diamond_{\underline{\leftrightarrow}_{ss}} = \frac{1}{1+\theta}(0, \theta, 1).$$

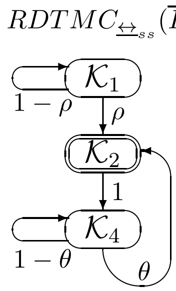

FIG. 9. The reduced quotient DTMC of $\overline{F}$ for $F = [(\{a\},\rho) * ((\{b\},\natural_k^1);((((\{c\},\natural_l^0);(\{d\},\theta)_1)[]((\{c\},\natural_m^0);(\{d\},\theta)_2))) * \mathsf{Stop}]$

Note that $\psi^\diamond_{\underline{\leftrightarrow}_{ss}} = (\psi^\diamond_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_1), \psi^\diamond_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_2), \psi^\diamond_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_4))$. *By the "quotient" analogue of Proposition 5 from* [87], *we have*

$$\begin{aligned} &\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_1) = 0,\\ &\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_2) = \tfrac{\theta}{1+\theta},\\ &\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_3) = 0,\\ &\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_4) = \tfrac{1}{1+\theta}. \end{aligned}$$

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

$$\varphi_{\underline{\leftrightarrow}_{ss}} = \frac{1}{1+\theta}(0,\theta,0,1).$$

This coincides with the result obtained in Example 8 with the use of $\psi^*_{\underline{\leftrightarrow}_{ss}}$ *and* $SJ_{\underline{\leftrightarrow}_{ss}}$.

Example 13. *Let F be from Example 6. In Figure 10, the reduced quotient SMC* $RSMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ *is depicted. The average sojourn times in the states of the reduced quotient SMC are written next to them in bold font. In spite of the equality* $RSMC_{\underline{\leftrightarrow}_{ss}}(\overline{F}) = RDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$, *the graphical representation of* $RSMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ *differs from that of* $RDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$, *since the former is based on the* $REDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$, *where each state is decorated with the* positive *average sojourn time of* $RSMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ *in it.* $REDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ *can be constructed from* $EDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ *in the similar way as* $RDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ *can be obtained from* $DTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$. *By construction, the residence time in each state of* $RSMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ *is geometrically distributed. Hence, the associated parameter of geometrical distribution is uniquely recovered from the average sojourn time in the state.*

The relationships between the steady-state PMFs $\psi_{\underline{\leftrightarrow}_{ss}}$ and $\psi^*_{\underline{\leftrightarrow}_{ss}}$, $\varphi_{\underline{\leftrightarrow}_{ss}}$ and $\psi_{\underline{\leftrightarrow}_{ss}}$, as well as $\varphi_{\underline{\leftrightarrow}_{ss}}$ and $\psi^\diamond_{\underline{\leftrightarrow}_{ss}}$, are the same as those between their "non-quotient" versions in Proposition 3, Proposition 4 and Proposition 5 from [87], respectively.

5.2. **Interrelations of the standard and quotient behavioural structures.** In Figure 11, the cube of interconnections by the relation "constructed from" is depicted for both the standard and quotient transition systems and Markov chains (SMCs, DTMCs and RDTMCs) of the process expressions. The relations between SMC and $SMC_{\underline{\leftrightarrow}_{ss}}$, between $DTMC$ and $DTMC_{\underline{\leftrightarrow}_{ss}}$, as well as between $RDTMC$

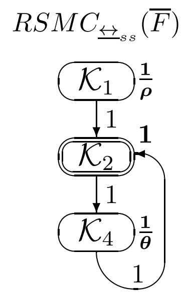

FIG. 10. The reduced quotient SMC of $\overline{F}$ for $F = [(\{a\},\rho) * ((\{b\},\natural_k^1);(((\{c\},\natural_l^0);(\{d\},\theta)_1)[]((\{c\},\natural_m^0);(\{d\},\theta)_2))) * \mathsf{Stop}]$

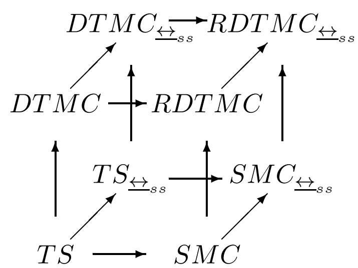

FIG. 11. The cube of interrelations for the standard and quotient transition systems and Markov chains of the process expressions

and $RDTMC_{\underline{\leftrightarrow}_{ss}}$, can be obtained using the following corresponding transition functions, defined by analogy with those already introduced: $PM^*(\mathcal{K},\widetilde{\mathcal{K}})$, based on $PM^*(s,\tilde{s})$, then $PM(\mathcal{K},\widetilde{\mathcal{K}})$, based on $PM(s,\tilde{s})$, as well as $PM^\diamond(\mathcal{K},\widetilde{\mathcal{K}})$, based on $PM^\diamond(s,\tilde{s})$ (all that to be proved below).

The relations between SMC and $RDTMC$, between $SMC_{\underline{\leftrightarrow}_{ss}}$ and $RDTMC_{\underline{\leftrightarrow}_{ss}}$, can be obtained using the next corresponding transition functions: $PM^\diamond(s,\tilde{s})$, based on $PM^*(s,\tilde{s})$, through $(PM^\diamond)^*(s,\tilde{s})$, as well as $PM^\diamond(\mathcal{K},\widetilde{\mathcal{K}})$, based on $PM^*(\mathcal{K},\widetilde{\mathcal{K}})$, through $(PM^\diamond)^*(\mathcal{K},\widetilde{\mathcal{K}})$ (by Theorem 5.2 from [88] and its "quotient" analogue).

In Figure 11, the relation (arrow) between $DTMC$ and $DTMC_{\underline{\leftrightarrow}_{ss}}$ is obtained using the transition function $PM(\mathcal{K},\widetilde{\mathcal{K}})$, based on $PM(s,\tilde{s})$. Let G be a dynamic expression. We shall prove that the (quotient) TPM $\mathbf{P}_{\underline{\leftrightarrow}_{ss}}$ for $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$, (forwardly) constructed by quotienting (by $\underline{\leftrightarrow}_{ss}$) $TS(G)$, followed by extracting $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$ from $TS_{\underline{\leftrightarrow}_{ss}}(G)$, *coincides with* the TPM $(\mathbf{P})_{\underline{\leftrightarrow}_{ss}}$, (reversely) constructed by extracting $DTMC(G)$ from $TS(G)$, followed by quotienting $DTMC(G)$. The next proposition relates those quotient extracted TPM $(\mathbf{P})_{\underline{\leftrightarrow}_{ss}}$ and extracted quotient TPM $\mathbf{P}_{\underline{\leftrightarrow}_{ss}}$.

Proposition 7. *Let G be a dynamic expression, $\mathbf{P}_{\underline{\leftrightarrow}_{ss}}$ be the TPM for $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$ and $(\mathbf{P})_{\underline{\leftrightarrow}_{ss}}$ results from quotienting (by $\underline{\leftrightarrow}_{ss}$) the TPM $\mathbf{P}$ for $DTMC(G)$. Then*

$$(\mathbf{P})_{\underline{\leftrightarrow}_{ss}} = \mathbf{P}_{\underline{\leftrightarrow}_{ss}}.$$

Proof. Let $\mathcal{K},\widetilde{\mathcal{K}} \in DR(G)/_{\mathcal{R}_{ss}(G)}$ and $s \in \mathcal{K}$.

In $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$, we have $\sum_{A\in\mathbb{N}_{fin}^{\mathcal{L}}} PM_A(\mathcal{K},\widetilde{\mathcal{K}}) = \sum_{A\in\mathbb{N}_{fin}^{\mathcal{L}}} PM_A(s,\widetilde{\mathcal{K}}) = \sum_{A\in\mathbb{N}_{fin}^{\mathcal{L}}} \sum_{\{\Upsilon\mid\exists\tilde{s}\in\widetilde{\mathcal{K}}\ s\stackrel{\Upsilon}{\to}\tilde{s},\ \mathcal{L}(\Upsilon)=A\}} PT(\Upsilon,s) = \sum_{\{\Upsilon\mid\exists\tilde{s}\in\widetilde{\mathcal{K}}\ s\stackrel{\Upsilon}{\to}\tilde{s}\}} PT(\Upsilon,s) = PM(s,\widetilde{\mathcal{K}}) = PM(\mathcal{K},\widetilde{\mathcal{K}})$.

In the quotient of $DTMC(G)$, we have $\sum_{\tilde{s}\in\widetilde{\mathcal{K}}} PM(s,\tilde{s}) = \sum_{\tilde{s}\in\widetilde{\mathcal{K}}} \sum_{\{\Upsilon\mid s\stackrel{\Upsilon}{\to}\tilde{s}\}} PT(\Upsilon,s) = \sum_{\{\Upsilon\mid\exists\tilde{s}\in\widetilde{\mathcal{K}}\ s\stackrel{\Upsilon}{\to}\tilde{s}\}} PT(\Upsilon,s) = PM(s,\widetilde{\mathcal{K}}) = PM(\mathcal{K},\widetilde{\mathcal{K}})$.

Thus, $(\mathbf{P})_{\underline{\leftrightarrow}_{ss}} = \mathbf{P}_{\underline{\leftrightarrow}_{ss}}$. □

Hence, the quotienting and extraction are permutable for transition systems of the process expressions. Applying extraction before the quotienting is useful to start from the level of Markov chains in the proofs.

Example 14. *Let F be from Example 4. The TPMs for $DTMC(\overline{F})$ and $DTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ are*

$$\mathbf{P} = \begin{pmatrix} 1-\rho & \rho & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{l}{l+m} & \frac{m}{l+m} \\ 0 & \theta & 0 & 1-\theta & 0 \\ 0 & \theta & 0 & 0 & 1-\theta \end{pmatrix},\ \mathbf{P}_{\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} 1-\rho & \rho & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \theta & 0 & 1-\theta \end{pmatrix}.$$

The TPM for the quotient of $DTMC(\overline{F})$ is

$$(\mathbf{P})_{\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} 1-\rho & \rho & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \theta & 0 & 1-\theta \end{pmatrix}.$$

Then it is clear that

$$(\mathbf{P})_{\underline{\leftrightarrow}_{ss}} = \mathbf{P}_{\underline{\leftrightarrow}_{ss}}.$$

In Figure 11, the relation (depicted by arrow) between SMC and $SMC_{\underline{\leftrightarrow}_{ss}}$ is obtained using the transition function $PM^*(\mathcal{K},\widetilde{\mathcal{K}})$, based on $PM^*(s,\tilde{s})$. Let G be a dynamic expression. We shall prove that the (quotient) TPM $\mathbf{P}^*_{\underline{\leftrightarrow}_{ss}}$ for $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$, (forwardly) constructed by quotienting (by $\underline{\leftrightarrow}_{ss}$) $DTMC(G)$, followed by embedding $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ into $SMC_{\underline{\leftrightarrow}_{ss}}(G)$, *coincides with* the (finally) embedded TPM $(\mathbf{P}^*)^*_{\underline{\leftrightarrow}_{ss}}$, (reversely) constructed by embedding $EDTMC(G)$ into $SMC(G)$, followed by quotienting $EDTMC(G)$, and final embedding a new EDTMC $EDTMC'(G)$ into the quotient of $EDTMC(G)$. The final embedding in the reverse construction is needed, since new self-loops may arise after quotienting $EDTMC(G)$, i.e. it may become not an EDTMC, but a DTMC featuring self-loops with probability less than 1. Note that for $\mathcal{K}\in DR(G)/_{\mathcal{R}_{ss}(G)}$ and $s\in\mathcal{K}$, we have $PM^*(\mathcal{K},\widetilde{\mathcal{K}}) = SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K})PM(\mathcal{K},\widetilde{\mathcal{K}}) = SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K})PM(s,\widetilde{\mathcal{K}})$ in $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$. This corresponds to a different expression $\sum_{\tilde{s}\in\widetilde{\mathcal{K}}} PM^*(s,\tilde{s}) = \sum_{\tilde{s}\in\widetilde{\mathcal{K}}} SL(s)PM(s,\tilde{s}) = SL(s)\sum_{\tilde{s}\in\widetilde{\mathcal{K}}} PM(s,\tilde{s}) = SL(s)PM(s,\widetilde{\mathcal{K}})$ in the quotient of $EDTMC(G)$. In particular, $SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) > SL(s)$ when $PM(s,\mathcal{K}\setminus\{s\}) > 0$, which is the reason for a new self-loop associated with s in the quotient of $EDTMC(G)$. The next proposition relates those finally embedded quotient embedded TPM $(\mathbf{P}^*)^*_{\underline{\leftrightarrow}_{ss}}$ (the TPM for $EDTMC'(G)$) and embedded quotient TPM $\mathbf{P}^*_{\underline{\leftrightarrow}_{ss}}$.

Proposition 8. *Let G be a dynamic expression, $\mathbf{P}^*_{\underline{\leftrightarrow}_{ss}}$ be the TPM for $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ and $(\mathbf{P}^*)^*_{\underline{\leftrightarrow}_{ss}}$ results from quotienting (by $\underline{\leftrightarrow}_{ss}$) and final embedding the TPM $\mathbf{P}^*$ for $EDTMC(G)$. Then*

$$(\mathbf{P}^*)^*_{\underline{\leftrightarrow}_{ss}} = \mathbf{P}^*_{\underline{\leftrightarrow}_{ss}}.$$

Proof. See Appendix A.1. □

Thus, the quotienting before embedding is more optimal computationally for DTMCs of the process expressions.

By Proposition 8, $EDTMC'(G) = EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$. The sojourn time in every $\mathcal{K} \in DR_T(G)/_{\mathcal{R}_{ss}(G)}$ is geometrically distributed with the parameter $\frac{1}{SL(s)SL'(s,\mathcal{K})} = \frac{1}{SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K})}$, where $SL'(s,\mathcal{K}) = \frac{1}{1-SL(s)PM(s,\mathcal{K}\setminus\{s\})}$, while the sojourn time in every $\mathcal{K}{\in}DR_V(G)/_{\mathcal{R}_{ss}(G)}$ is equal to 0. Here $SL'(s,\mathcal{K})$ is the self-loops abstraction factor in the equivalence class $\mathcal{K}$ with respect to the state $s \in \mathcal{K}$ for the quotient of $EDTMC(G)$. Hence, $SMC'(G){=}SMC_{\underline{\leftrightarrow}_{ss}}(G)$, where $SMC'(G)$ is the SMC with the EDTMC $EDTMC'(G)$, such that $\frac{1}{SL(s)SL'(s,\mathcal{K})}$ is the geometrical distribution parameter of the sojourn time in every $\mathcal{K} \in DR_T(G)/_{\mathcal{R}_{ss}(G)}$ while the sojourn time is zero in every $\mathcal{K} \in DR_V(G)/_{\mathcal{R}_{ss}(G)}$.

Example 15. *Let F be from Example 4. The TPMs for $EDTMC(\overline{F})$ and $EDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ are*

$$\mathbf{P}^* = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{l}{l+m} & \frac{m}{l+m} \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{P}^*_{\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

The TPMs for the quotient of $EDTMC(\overline{F})$ and EDTMC of the quotient of $EDTMC(\overline{F})$ ($EDTMC'(\overline{F})$), are

$$(\mathbf{P}^*)_{\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad (\mathbf{P}^*)^*_{\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Then it is clear that

$$(\mathbf{P}^*)^*_{\underline{\leftrightarrow}_{ss}} = \mathbf{P}^*_{\underline{\leftrightarrow}_{ss}}.$$

Let G be a dynamic expression. We now construct the quotient (by $\underline{\leftrightarrow}_{ss}$) of the TPM for $DTMC(G)$ using special collector and distributor matrices. Let $DR(G) = \{s_1, \ldots, s_n\}$ and $DR(G)/_{\mathcal{R}_{ss}(G)} = \{\mathcal{K}_1, \ldots, \mathcal{K}_l\}$.

The elements $(\mathcal{P}_{\underline{\leftrightarrow}_{ss}})_{rs}$ $(1 \le r, s \le l)$ of the TPM $\mathbf{P}_{\underline{\leftrightarrow}_{ss}}$ for $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$ are defined as

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

Like it has been done for strong performance bisimulation on labeled CTSPNs in [31], the $l \times l$ TPM $\mathbf{P}_{\underline{\leftrightarrow}_{ss}}$ for $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$ can be constructed from the $n \times n$ TPM

$\mathbf{P}$ for $DTMC(G)$ using the $n \times l$ *collector matrix* $\mathbf{V}$ for the largest step stochastic autobisimulation $\mathcal{R}_{ss}(G)$ on G and the $l \times n$ *distributor matrix* $\mathbf{W}$ for $\mathbf{V}$. Then $\mathbf{W}$ should be a non-negative matrix (i.e. all its elements must be non-negative) with the elements of each its row summed to one, such that $\mathbf{WV} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix of order l, i.e. $\mathbf{W}$ is a *left-inverse matrix* for $\mathbf{V}$. It is known that for each collector matrix there is at least one distributor matrix, in particular, the matrix obtained by transposing $\mathbf{V}$ and subsequent normalizing its rows, to guarantee that the elements of each row of the transposed matrix are summed to one. We now present the formal definitions.

The elements $\mathcal{V}_{ir}$ $(1 \le i \le n,\ 1 \le r \le l)$ of the *collector matrix* $\mathbf{V}$ for the largest step stochastic autobisimulation $\mathcal{R}_{ss}(G)$ on G are defined as

$$\mathcal{V}_{ir} = \begin{cases} 1, & s_i \in \mathcal{K}_r; \\ 0, & \text{otherwise.} \end{cases}$$

Thus, all the elements of $\mathbf{V}$ are non-negative, as required. The row elements of $\mathbf{V}$ are summed to one, since for each s_i $(1 \le i \le n)$ there exists exactly one $\mathcal{K}_r$ $(1 \le r \le l)$ such that $s_i \in \mathcal{K}_r$. Hence,

$$\mathbf{V}\mathbf{1}^T = \mathbf{1}^T,$$

where $\mathbf{1}$ on the left side is the row vector of l values 1 while $\mathbf{1}$ on the right side is the row vector of n values 1.

The *distributor matrix* $\mathbf{W}$ for the collector matrix $\mathbf{V}$ is defined as

$$\mathbf{W} = (Diag(\mathbf{V}^T\mathbf{1}^T))^{-1}\mathbf{V}^T,$$

where $\mathbf{1}$ is the row vector of n values 1. One can check that $\mathbf{WV} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix of order l.

The elements $(\mathcal{PV})_{is}$ $(1 \le i \le n,\ 1 \le s \le l)$ of the matrix $\mathbf{PV}$ are

$$(\mathcal{PV})_{is} = \sum_{j=1}^{n} \mathcal{P}_{ij}\mathcal{V}_{js} = \sum_{\{j \mid 1 \le j \le n,\ s_j \in \mathcal{K}_s\}} PM(s_i, s_j) = PM(s_i, \mathcal{K}_s).$$

For each s_i $(1 \le i \le n)$ there exists exactly one $\mathcal{K}_r$ $(1 \le r \le l)$ such that $s_i \in \mathcal{K}_r$. For all $s_i \in \mathcal{K}_r$ we have $PM(\mathcal{K}_r, \mathcal{K}_s) = PM(s_i, \mathcal{K}_s)$ $(1 \le i \le n,\ 1 \le r, s \le l)$. Then the elements $(\mathcal{VP}_{\underline{\leftrightarrow}_{ss}})_{is}$ $(1 \le i \le n,\ 1 \le s \le l)$ of the matrix $\mathbf{VP}_{\underline{\leftrightarrow}_{ss}}$ are

$$(\mathcal{VP}_{\underline{\leftrightarrow}_{ss}})_{is} = \sum_{r=1}^{l} \mathcal{V}_{ir}(\mathcal{P}_{\underline{\leftrightarrow}_{ss}})_{rs} = \sum_{\{r \mid 1 \le r \le l,\ s_i \in \mathcal{K}_r\}} PM(\mathcal{K}_r, \mathcal{K}_s) = PM(s_i, \mathcal{K}_s).$$

Therefore, we have

$$\mathbf{PV} = \mathbf{VP}_{\underline{\leftrightarrow}_{ss}},\ \mathbf{WPV} = \mathbf{P}_{\underline{\leftrightarrow}_{ss}}.$$

Example 16. *Let F be from Example 4. The TPMs for $DTMC(\overline{F})$ and $DTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ are*

$$\mathbf{P} = \begin{pmatrix} 1-\rho & \rho & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{l}{l+m} & \frac{m}{l+m} \\ 0 & \theta & 0 & 1-\theta & 0 \\ 0 & \theta & 0 & 0 & 1-\theta \end{pmatrix}, \ \mathbf{P}_{\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} 1-\rho & \rho & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \theta & 0 & 1-\theta \end{pmatrix}.$$

The collector matrix $\mathbf{V}$ *for* $\mathcal{R}_{ss}(\overline{F})$ *and the distributor matrix* $\mathbf{W}$ *for* $\mathbf{V}$ *are*

$$\mathbf{V} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ \mathbf{W} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

Then it is easy to check that

$$\mathbf{WPV} = \mathbf{P}_{\underline{\leftrightarrow}_{ss}}.$$

In Figure 11, the relation (arrow) between $RDTMC$ and $RDTMC_{\underline{\leftrightarrow}_{ss}}$ is obtained using the transition function $PM^{\diamond}(\mathcal{K}, \widetilde{\mathcal{K}})$, based on $PM^{\diamond}(s, \tilde{s})$. Let G be a dynamic expression. We shall prove that the TPM $\mathbf{P}^{\diamond}_{\underline{\leftrightarrow}_{ss}}$, (forwardly) constructed by quotienting (by $\underline{\leftrightarrow}_{ss}$) $DTMC(G)$, followed by reduction (eliminating vanishing states) of $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$, *coincides with* the TPM $(\mathbf{P}^{\diamond})_{\underline{\leftrightarrow}_{ss}}$, (reversely) constructed by reduction of $DTMC(G)$, followed by quotienting $RDTMC(G)$. The next proposition relates those quotient reduced TPM $(\mathbf{P}^{\diamond})_{\underline{\leftrightarrow}_{ss}}$ and reduced quotient TPM $\mathbf{P}^{\diamond}_{\underline{\leftrightarrow}_{ss}}$.

Proposition 9. *Let* G *be a dynamic expression,* $\mathbf{P}^{\diamond}_{\underline{\leftrightarrow}_{ss}}$ *be the TPM for* $RDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ *and* $(\mathbf{P}^{\diamond})_{\underline{\leftrightarrow}_{ss}}$ *results from quotienting (by* $\underline{\leftrightarrow}_{ss}$*) the TPM* $\mathbf{P}^{\diamond}$ *for* $RDTMC(G)$*. Then*

$$(\mathbf{P}^{\diamond})_{\underline{\leftrightarrow}_{ss}} = \mathbf{P}^{\diamond}_{\underline{\leftrightarrow}_{ss}}.$$

Proof. See Appendix A.2. □

Thus, the quotienting and reduction are permutable for DTMCs of the process expressions. This may simplify the performance evaluation when eliminating vanishing states makes the subsequent quotienting more efficient. The reverse construction (reduction first) is particularly preferable in case of small equivalence classes of vanishing states when quotienting does not merge many of them before eliminating.

Example 17. *Let* F *be from Example 4. The reordered TPMs for* $DTMC(\overline{F})$ *and* $DTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ *are*

$$\mathbf{P}_r = \begin{pmatrix} 0 & 0 & 0 & \frac{l}{l+m} & \frac{m}{l+m} \\ 0 & 1-\rho & \rho & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \theta & 1-\theta & 0 \\ 0 & 0 & \theta & 0 & 1-\theta \end{pmatrix}, \ \mathbf{P}_{r\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1-\rho & \rho & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & \theta & 1-\theta \end{pmatrix}.$$

The reordered collector matrix $\mathbf{V}_r$ *for* $\mathcal{R}_{ss}(\overline{F})$ *and the reordered distributor matrix* $\mathbf{W}_r$ *for* $\mathbf{V}_r$ *are*

$$\mathbf{V}_r = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \ \mathbf{W}_r = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

Then it is easy to check that

$$\mathbf{W}_r\mathbf{P}_r\mathbf{V}_r = \mathbf{P}_{r\underline{\leftrightarrow}_{ss}}.$$

Example 18. *Let F be from Example 4. The TPMs for $RDTMC(\overline{F})$ and $RDTMC_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ are*

$$\mathbf{P}^{\diamond} = \begin{pmatrix} 1-\rho & \rho & 0 & 0 \\ 0 & 0 & \frac{l}{l+m} & \frac{m}{l+m} \\ 0 & \theta & 1-\theta & 0 \\ 0 & \theta & 0 & 1-\theta \end{pmatrix}, \ \mathbf{P}^{\diamond}_{\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} 1-\rho & \rho & 0 \\ 0 & 0 & 1 \\ 0 & \theta & 1-\theta \end{pmatrix}.$$

The result of the decomposing the reordered collector matrix $\mathbf{V}_r$ for $\mathcal{R}_{ss}(\overline{F})$ and the reordered distributor matrix $\mathbf{W}_r$ for $\mathbf{V}_r$ are the matrices

$$\mathbf{V}_C = 1, \ \mathbf{V}_F = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \ \mathbf{W}_C = 1, \ \mathbf{W}_F = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

Then it is easy to check that

$$(\mathbf{P}^{\diamond})_{\underline{\leftrightarrow}_{ss}} = \mathbf{W}_F\mathbf{P}^{\diamond}\mathbf{V}_F = \mathbf{P}^{\diamond}_{\underline{\leftrightarrow}_{ss}}.$$

In [30], the ordinary, exact and strict lumpability relations on finite DTMCs are explored. It is investigated which properties of transient and stationary behaviour of DTMCs are preserved by aggregation with respect to the three mentioned kinds of lumping and their approximate "nearly" versions. It is proved that irreducibility is preserved by aggregation with respect to any partition (or equivalence relation) on the states of DTMCs. Since only finite irreducible DTMCs are considered (with a finite number of states), these all are positive recurrent. Aggregation can only decrease the number of states, hence, the aggregated DTMCs are also finite and positive recurrence is preserved by every aggregation. It is known [77, 80, 52, 26, 95, 53, 78, 79] that irreducible and positive recurrent DTMCs have a single stationary PMF. Note that the original and aggregated DTMCs may be periodic, thus having a unique stationary distribution, but no steady-state (limiting) one. For example, it may happen that the original DTMC is aperiodic while the aggregated DTMC is periodic due to merging some states of the former. Thus, both finite irreducible DTMCs and their arbitrary aggregates have a single stationary PMF. Then the relationship between stationary probabilities of DTMCs and their aggregates with respect to ordinary, exact and strict lumpability is established in [30]. In particular, it is shown that for every DTMC aggregated by ordinary lumpability, the stationary probability of each aggregate state is a sum of the stationary probabilities of all its constituent states from the original DTMC. The information about individual stationary probabilities of the original DTMC is lost after such a summation, but in many cases, the stationary probabilities of the aggregated DTMC are enough to calculate performance measures of the high-level model, from which the original DTMC is extracted. As mentioned in [30], in some practical applications, the aggregated DTMC can be extracted directly from the high-level model. Thus, the aggregation techniques based on lumping are of practical importance, since they allow one to reduce the state space of the modeled systems, hence, the computational costs for evaluating their performance.

Let G be a dynamic expression. By definition of $\underline{\leftrightarrow}_{ss}$, the relation $\mathcal{R}_{ss}(G)$ on $TS(G)$ induces ordinary lumping on $SMC(G)$, i.e. if the states of $TS(G)$ are related by $\mathcal{R}_{ss}(G)$ then the same states in $SMC(G)$ are related by ordinary lumping. The quotient (maximal aggregate) of $SMC(G)$ by such an induced ordinary lumping is $SMC_{\underline{\leftrightarrow}_{ss}}(G)$. Since we consider only finite SMCs, irreducibility of $SMC(G)$ will imply irreducibility of $SMC_{\underline{\leftrightarrow}_{ss}}(G)$ and they both are positive recurrent. Then a unique quotient stationary PMF of $SMC_{\underline{\leftrightarrow}_{ss}}(G)$ can be calculated from a unique original stationary PMF of $SMC(G)$ by summing some elements of the latter, as described in [30]. Similar arguments demonstrate that the same results hold for $DTMC(G)$ and $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$, as well as for $RDTMC(G)$ and $RDTMC_{\underline{\leftrightarrow}_{ss}}(G)$.

6. Stationary behaviour

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

Like in the framework of SMCs, in LDTSDPNs the most common systems for performance analysis are *ergodic* (irreducible, positive recurrent and aperiodic) ones. For ergodic LDTSDPNs, the steady-state marking probabilities exist and can be determined. In [70, 71], the following sufficient (but not necessary) conditions for ergodicity of DTSPNs are stated: *liveness* (for each transition and any reachable marking there exists a sequence of markings from it leading to the marking enabling that transition), *boundedness* (for any reachable marking the number of tokens in every place is not greater than some fixed number) and *nondeterminism* (the transition probabilities are strictly less than 1). However, it has been shown in [7] that even live, safe and nondeterministic DTSPNs (as well as live and safe CTSPNs and GSPNs) may be non-ergodic.

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

Proposition 10. *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. The standard proof is analogous to that of Proposition 6 from [93]. For the alternative proof, see Appendix A.3. □

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 10 (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 $\mathcal{K}$.

Let $\mathbf{V}$ be the collector matrix for $\mathcal{R}_{ss}(G)$. One can see that

$$\varphi \mathbf{V} = \varphi_{\underline{\leftrightarrow}_{ss}}.$$

Hence, using $SMC_{\underline{\leftrightarrow}_{ss}}(G)$ instead of $SMC(G)$ may simplify the analytical solution, since we may have less states, but constructing the TPM for $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$, denoted by $\mathbf{P}^*_{\underline{\leftrightarrow}_{ss}}$, also requires some efforts, including determining $\mathcal{R}_{ss}(G)$ and calculating the probabilities to move from one equivalence class to other. The behaviour of $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$ may stabilize quicker than that of $EDTMC(G)$ (if each of them has a single steady state), since $\mathbf{P}^*_{\underline{\leftrightarrow}_{ss}}$ is generally denser matrix than $\mathbf{P}^*$ (the TPM for $EDTMC(G)$), since the former matrix is usually smaller and the transitions between the equivalence classes "include" all the transitions between the states belonging to these equivalence classes.

By Proposition 10, $\underline{\leftrightarrow}_{ss}$ preserves the quantitative properties of the stationary behaviour (the level of SMCs). We now intend to demonstrate that the qualitative properties of the stationary behaviour based on the multiaction labels are preserved as well (the level of transition systems).

Definition 22. *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 $\exists s \in DR(G)\ s \stackrel{\Upsilon_1}{\to} s_1 \stackrel{\Upsilon_2}{\to} \cdots \stackrel{\Upsilon_n}{\to} s_n,\ \mathcal{L}(\Upsilon_i) = A_i\ (1 \le i \le n)$. Then the* probability to execute the derived step trace Σ in s *is*

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

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 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. The proof is analogous to that of Theorem 4 from [93]. □

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

Let $DR(G) = \{s_1, \dots, s_n\}$ and $DR(G)/_{\mathcal{R}_{ss}(G)} = \{\mathcal{K}_1, \dots, \mathcal{K}_l\}$ while $\mathbf{V}$ be the collector matrix for $\mathcal{R}_{ss}(G)$ and $\mathbf{W}$ be the distributor matrix for $\mathbf{V}$. We denote $PT(\Sigma) = (PT(\Sigma, s_1), \dots, PT(\Sigma, s_n))$ and $PT_{\underline{\leftrightarrow}_{ss}}(\Sigma) = (PT(\Sigma, \mathcal{K}_1), \dots, PT(\Sigma, \mathcal{K}_l))$. One can see that $Diag(PT(\Sigma))\mathbf{V} = \mathbf{V}Diag(PT_{\underline{\leftrightarrow}_{ss}}(\Sigma))$ and $\mathbf{W}Diag(PT(\Sigma))\mathbf{V} = Diag(PT_{\underline{\leftrightarrow}_{ss}}(\Sigma))$. Then we have

$$\varphi Diag(PT(\Sigma))\mathbf{V} = \varphi\mathbf{V}Diag(PT_{\underline{\leftrightarrow}_{ss}}(\Sigma)) = \varphi_{\underline{\leftrightarrow}_{ss}} Diag(PT_{\underline{\leftrightarrow}_{ss}}(\Sigma)).$$

We now present a result that does not concern the steady-state probabilities, but it reveals two very important properties of residence time in the equivalence classes. 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 11. *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. The proof is analogous to that of Proposition 7 from [93]. □

Example 19. *Let F be from Example 6. Consider the equivalence class (with respect to $\mathcal{R}_{ss}(\overline{F})$) $\mathcal{K}_4 = \{s_4, s_5\}$. Then the value of $\varphi_{\underline{\leftrightarrow}_{ss}}$ corresponding to $\mathcal{K}_4$ is the sum of all values of φ corresponding to the states from $\mathcal{K}_4$: $\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_4) = \frac{1}{1+\theta} = \frac{l}{(l+m)(1+\theta)} + \frac{m}{(l+m)(1+\theta)} = \varphi(s_4) + \varphi(s_5) = \sum_{s\in\mathcal{K}_4} \varphi(s)$.*

Let $\Sigma = \{\{d\}\}$. We have $\varphi_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_4)PT(\Sigma, \mathcal{K}_4) = \frac{1}{1+\theta} \cdot \theta = \frac{\theta}{1+\theta} = \frac{l}{(l+m)(1+\theta)} \cdot \theta + \frac{m}{(l+m)(1+\theta)} \cdot \theta = \varphi(s_4)PT(\{(\{d\}, \theta)_1\}, s_4) + \varphi(s_5)PT(\{(\{d\}, \theta)_2\}, s_5) = \varphi(s_4)PT(\Sigma, s_4) + \varphi(s_5)PT(\Sigma, s_5) = \sum_{s\in\mathcal{K}_4} \varphi(s)PT(\Sigma, s)$, where $PT(\Sigma, \mathcal{K}_4) = PT(\Sigma, s_4) = PT(\Sigma, s_5) = \theta$.

The sojourn time average in $\mathcal{K}_4$ *is* $SJ_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_4) = \frac{1}{1-PM(\mathcal{K}_4,\mathcal{K}_4)} = \frac{1}{1-(1-\theta)} = \frac{1}{\theta} = \frac{1}{1-(1-\theta)} = \frac{1}{1-PM(s_4,s_4)} = \frac{1}{1-PM(s_5,s_5)} = \frac{1}{1-PM(\{s_4,s_5\},\{s_4,s_5\})} = SJ_{\underline{\leftrightarrow}_{ss}}(\{s_4,s_5\})$.

The sojourn time variance in $\mathcal{K}_4$ *is* $VAR_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}_4) = \frac{PM(\mathcal{K}_4,\mathcal{K}_4)}{(1-PM(\mathcal{K}_4,\mathcal{K}_4))^2} = \frac{1-\theta}{(1-(1-\theta))^2} = \frac{1-\theta}{\theta^2} = \frac{1-\theta}{(1-(1-\theta))^2} = \frac{PM(s_4,s_4)}{(1-PM(s_4,s_4))^2} = \frac{PM(s_5,s_5)}{(1-PM(s_5,s_5))^2} = \frac{PM(\{s_4,s_5\}}{(1-PM(\{s_4,s_5\},\{s_4,s_5\}))^2} = VAR_{\underline{\leftrightarrow}_{ss}}(\{s_4,s_5\})$.

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 10, Theorem 3 and Proposition 11 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) After treating the transition system for self-similarity, a step stochastic autobisimulation equivalence for the expression is determined.
(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 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.

Before calculating stationary probabilities, we can further reduce the quotient underlying SMC, using an analogue of the deterministic barrier partitioning method described in [42] for semi-Markov processes (SMPs), which allows one to perform quicker the first passage-time analysis. Another option is the method of stochastic state classes proposed in [48, 49] for generalized SMPs (GSMPs) reduction, which allows one to simplify transient performance analysis (the analysis based on the transient probabilities of being in the states of GSMPs).

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 (instead

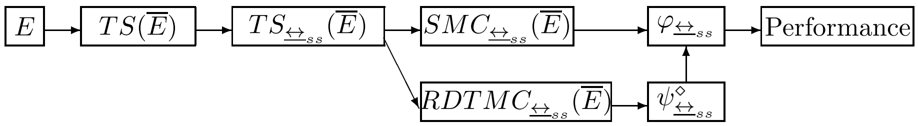

FIG. 12. Equivalence-based simplification of performance evaluation

of the quotient underlying SMC) 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 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.

We first merge the equivalent states in transition systems and then eliminate the vanishing states in Markov chains. The reason is that transition systems, being a higher-level formalism than Markov chains, describe both functional (qualitative) and performance (quantitative) aspects of behaviour while Markov chains represent only performance ones. Eliminating vanishing states first would destroy the functional behaviour (respected by the equivalence used for quotienting), since the steps with different multiaction parts may lead to or start from different vanishing states.

Figure 12 presents the main stages of the standard and alternative equivalence-based simplification of performance evaluation described above.

7. CONCLUSION

In this paper, we have considered dtsdPBC, an extension with discrete stochastic and deterministic time of Petri box calculus (PBC) [21, 23, 22]. Stochastic process algebra dtsdPBC has a parallel step operational semantics, based on labeled probabilistic transition systems, and a Petri net denotational semantics in terms of dtsd-boxes, a special subclass of LDTSDPNs [86]. Step stochastic bisimulation equivalence of the process expressions has been used to reduce their transition systems and Markov chains (SMCs, DTMCs and RDTMCs) with the quotienting. We have established isomorphism between the quotient transition systems of the process expressions and quotient reachability graphs of the corresponding dtsd-boxes, as well as between the quotient SMCs of the process expressions and quotient SMCs of the corresponding dtsd-boxes.

We have studied an effect of the quotienting to extraction, embedding and reduction, in terms of the transition probability matrices (TPMs) of the quotient DTMCs, EDTMCs and RDTMCs. We have demonstrated that for DTMCs of the process expressions, the quotienting is permutable (commute) with both extraction and reduction, whereas an additional embedding of the quotient embedded DTMC is needed to coincide with the embedded quotient DTMC. Thus, making extraction before the quotienting permits to start reasoning from the Markov chain level. Applying reduction before the quotienting simplifies quantitative analysis with many non-equivalent vanishing states. The quotienting before embedding diminishes computations.

We have proved that the mentioned 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. Therefore, quotienting by that *performance preserving equivalence* makes easier both the qualitative (functional) and quantitative (performance) analysis within dtsdPBC. Thus, we have outlined in dtsdPBC a novel method of modeling (system specification by a process expression and construction of its transition system), equivalence reduction (quotienting the transition system and possible elimination of vanishing states in the derived quotient SMC or DTMC) and simplified performance evaluation (calculation of the performance indices using the quotient SMC, DTMC or RDTMC). The advantage of the dtsdPBC framework is that the semantic parallelism level exhibited by the transition systems is maintained in the extracted performance models (SMCs, DTMCs and RDTMCs) through the state changes corresponding to the simultaneous executions.

Our method can be suitably applied to the stochastically and deterministically timed concurrent systems that adapt a discrete time concept. The examples of such systems are many industrial, manufacturing, queueing, computing and network systems with fixed durations of the typical activities and stochastic durations of the randomly occurring activities. Further examples include business processes, neural and transportation networks, computer and communication systems and timed web services [96] with discrete time, as well as highly distributed or massively parallel systems, such as genetic regulatory and cellular signalling networks in biology [37, 25, 6]. In [40], biological networks were jointly modeled by (standard, qualitative) PNs, CTSPNs and continuous PNs (CPNs), to demonstrate their complementarity that makes necessary adding deterministic time to stochastic models, as well as combining stochastic and continuous (deterministic) aspects into one model (such as stochastic rates of reactions and continuous amounts of species).

In future, we plan to construct the case studies demonstrating expressiveness of the calculus and application of the functional analysis and performance evaluation, both simplified using quotienting by step stochastic bisimulation. Future work could also consist in constructing a congruence relation for dtsdPBC, i.e. the equivalence that withstands application of all operations of the algebra. The first possible candidate is a stronger version of the equivalence with respect to transition systems, with two extra transitions skip and redo, like in sPBC [57]. Moreover, recursion operation could be added to dtsdPBC to increase specification power of the algebra.

Appendix A. Proofs

A.1. **Proof of Proposition 8.** Let $\mathcal{K},\widetilde{\mathcal{K}}\in DR(G)/_{\mathcal{R}_{ss}(G)}$ and $s\in\mathcal{K}$. The EDTMC for the quotient of $EDTMC(G)$ is denoted by $EDTMC'(G)$ and has the probabilities $PM'(\mathcal{K},\widetilde{\mathcal{K}})$ to change from $\mathcal{K}$ to $\widetilde{\mathcal{K}}$.

- Let $PM(s,s)+PM(s,\mathcal{K}\setminus\{s\})=PM(s,\mathcal{K})<1$ and $PM(s,s),PM(s,\mathcal{K}\setminus\{s\})>0$, i.e. $s,\mathcal{K}$ are non-absorbing and there exist self-loops associated with s in $DTMC(G)$ and with $\mathcal{K}$ in the quotient of $EDTMC(G)$.

  In $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$, we have $PM^*(\mathcal{K},\widetilde{\mathcal{K}})=SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K})PM(\mathcal{K},\widetilde{\mathcal{K}})=\frac{PM(\mathcal{K},\widetilde{\mathcal{K}})}{1-PM(\mathcal{K},\mathcal{K})}=\frac{PM(s,\widetilde{\mathcal{K}})}{1-PM(s,\mathcal{K})}=\frac{PM(s,\widetilde{\mathcal{K}})}{1-PM(s,s)-PM(s,\mathcal{K}\setminus\{s\})}=\frac{\frac{PM(s,\widetilde{\mathcal{K}})}{1-PM(s,s)}}{1-\frac{PM(s,\mathcal{K}\setminus\{s\})}{1-PM(s,s)}}=\frac{SL(s)PM(s,\widetilde{\mathcal{K}})}{1-SL(s)PM(s,\mathcal{K}\setminus\{s\})}$. Then $SL_{\underline{\leftrightarrow}_{ss}}(\mathcal{K})=\frac{SL(s)}{1-SL(s)PM(s,\mathcal{K}\setminus\{s\})}=$

$SL(s)SL'(s,\mathcal{K})$, where $SL'(s,\mathcal{K}) = \frac{1}{1-SL(s)PM(s,\mathcal{K}\setminus\{s\})}$ is the self-loops abstraction factor in the equivalence class $\mathcal{K}$ with respect to the state $s \in \mathcal{K}$ for the quotient of $EDTMC(G)$.

In $EDTMC'(G)$, we have $PM'(\mathcal{K},\widetilde{\mathcal{K}}) = \frac{\sum_{\tilde{s}\in\widetilde{\mathcal{K}}} PM^*(s,\tilde{s})}{1-\sum_{s'\in\mathcal{K}\setminus\{s\}} PM^*(s,s')} = \frac{\sum_{\tilde{s}\in\widetilde{\mathcal{K}}} SL(s)PM(s,\tilde{s})}{1-\sum_{s'\in\mathcal{K}\setminus\{s\}} SL(s)PM(s,s')} = \frac{SL(s)\sum_{\tilde{s}\in\widetilde{\mathcal{K}}} PM(s,\tilde{s})}{1-SL(s)\sum_{s'\in\mathcal{K}\setminus\{s\}} PM(s,s')} = \frac{SL(s)PM(s,\widetilde{\mathcal{K}})}{1-SL(s)PM(s,\mathcal{K}\setminus\{s\})} = PM^*(\mathcal{K},\widetilde{\mathcal{K}})$.

The other three cases (no self-loops associated with s in $DTMC(G)$, with $\mathcal{K}$ in the quotient of $EDTMC(G)$, or with both) are treated analogously, by replacing $PM(s,s)$ or/and $PM(s,\mathcal{K}\setminus\{s\})$ with zeros.

- Let $PM(s,s)+PM(s,\mathcal{K}\setminus\{s\}) = PM(s,\mathcal{K}) = 1$ and $PM(s,s), PM(s,\mathcal{K}\setminus\{s\}) > 0$, i.e. $\mathcal{K}$ is absorbing in $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$ and there exist self-loops associated with s in $DTMC(G)$ and with $\mathcal{K}$ in the quotient of $EDTMC(G)$.

  In $EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$, we have $PM^*(\mathcal{K},\mathcal{K}) = 1$ by definition of the EDTMC, since $PM(\mathcal{K},\mathcal{K}) = PM(s,\mathcal{K}) = 1$.

  In the quotient of $EDTMC(G)$, the probability of a self-loop associated with $\mathcal{K}$ is $\sum_{s'\in\mathcal{K}\setminus\{s\}} PM^*(s,s') = \sum_{s'\in\mathcal{K}\setminus\{s\}} SL(s)PM(s,s') = SL(s)\sum_{s'\in\mathcal{K}\setminus\{s\}} PM(s,s') = SL(s)PM(s,\mathcal{K}\setminus\{s\}) = SL(s)(1-PM(s,s)) = \frac{1-PM(s,s)}{1-PM(s,s)} = 1$. In $EDTMC'(G)$, we have $PM'(\mathcal{K},\widetilde{\mathcal{K}}) = 1 = PM^*(\mathcal{K},\mathcal{K})$ by definition of the EDTMC, since in the quotient of $EDTMC(G)$, the probability of a self-loop associated with $\mathcal{K}$ is 1.

  The other two cases (no self-loops associated with s in $DTMC(G)$ or with $\mathcal{K}$ in the quotient of $EDTMC(G)$) are treated analogously, by replacing $PM(s,s)$ with zero or taking $\mathcal{K} = \{s\}$ when $PM(s,\mathcal{K}\setminus\{s\}) = 0$.

Thus, $(\mathbf{P}^*)^*_{\underline{\leftrightarrow}_{ss}} = \mathbf{P}^*_{\underline{\leftrightarrow}_{ss}}$ and $EDTMC'(G) = EDTMC_{\underline{\leftrightarrow}_{ss}}(G)$. □

A.2. **Proof of Proposition 9.** Let $\mathbf{P}_r$ be the reordered (by moving vanishing states to the first positions) TPM for $DTMC(G)$. Like in [87], we reorder the states from $DR(G)$ so that the first rows and columns of $\mathbf{P}_r$ will correspond to the states from $DR_V(G)$ and the last ones will correspond to the states from $DR_T(G)$. Let $|DR(G)| = n$ and $|DR_T(G)| = m$. Then the reordered TPM for $DTMC(G)$ can be decomposed as

$$\mathbf{P}_r = \begin{pmatrix} \mathbf{C} & \mathbf{D} \\ \mathbf{E} & \mathbf{F} \end{pmatrix}.$$

The elements of the $(n-m)\times(n-m)$ submatrix $\mathbf{C}$ are the probabilities to move from vanishing to vanishing states, and those of the $(n-m)\times m$ submatrix $\mathbf{D}$ are the probabilities to move from vanishing to tangible states. The elements of the $m\times(n-m)$ submatrix $\mathbf{E}$ are the probabilities to move from tangible to vanishing states, and those of the $m\times m$ submatrix $\mathbf{F}$ are the probabilities to move from tangible to tangible states.

The TPM $\mathbf{P}^\diamond$ for $RDTMC(G)$ is the $m\times m$ matrix, calculated as

$$\mathbf{P}^\diamond = \mathbf{F} + \mathbf{EGD},$$

where the elements of the matrix $\mathbf{G} = \sum_{k=0}^{\infty} \mathbf{C}^k$ are the probabilities to move from vanishing to vanishing states in any number of state changes, without traversal of tangible states.

By the note after Proposition 3, $\mathcal{R}_{ss}(G) \subseteq (DR_T(G))^2 \uplus (DR_V(G))^2$. Hence, $\forall \mathcal{K} \in DR(G)/_{\mathcal{R}_{ss}(G)}$, all states from $\mathcal{K}$ are tangible, when $\mathcal{K} \in DR_T(G)/_{\mathcal{R}_{ss}(G)}$, or all of them are vanishing, when $\mathcal{K} \in DR_V(G)/_{\mathcal{R}}$.

Let $\mathbf{V}_r$ be the reordered (by moving vanishing states and their equivalence classes to the first positions) collector matrix for $\mathcal{R}_{ss}(\overline{F})$ and $\mathbf{W}_r$ be the (accordingly) reordered distributor matrix for $\mathbf{V}_r$. We reorder the states from $DR(G)$ and the equivalence classes from $DR(G)/_{\mathcal{R}_{ss}(G)}$ as follows. The first rows of $\mathbf{V}_r$ will correspond to the states from $DR_V(G)$ and the first columns of $\mathbf{V}_r$ will correspond to the equivalence classes from $DR_V(G)/_{\mathcal{R}_{ss}(G)}$, whereas the last rows of $\mathbf{V}_r$ will correspond to the states from $DR_T(G)$ and the last columns of $\mathbf{V}_r$ will correspond to the equivalence classes from $DR_T(G)/_{\mathcal{R}_{ss}(G)}$. The first rows of $\mathbf{W}_r$ will correspond to the equivalence classes from $DR_V(G)/_{\mathcal{R}_{ss}(G)}$ and the first columns of $\mathbf{W}_r$ will correspond to the states from $DR_V(G)$, whereas the last rows of $\mathbf{W}_r$ will correspond to the equivalence classes from $DR_T(G)/_{\mathcal{R}_{ss}(G)}$ and the last columns of $\mathbf{W}_r$ will correspond to the states from $DR_T(G)$.

Let $|DR(G)/_{\mathcal{R}_{ss}(G)}| = l$ and $|DR_T(G)/_{\mathcal{R}_{ss}(G)}| = k$. Note that tangible (vanishing) states can only belong to the equivalence classes of tangible (vanishing) states. Then the reordered collector and distributor matrices can be decomposed as

$$\mathbf{V}_r = \begin{pmatrix} \mathbf{V}_C & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_F \end{pmatrix}, \ \mathbf{W}_r = \begin{pmatrix} \mathbf{W}_C & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_F \end{pmatrix},$$

where $\mathbf{0}$ are the matrices consisting only of zeros, all those matrices of the appropriate sizes. The elements of the $(n-m)\times(l-k)$ submatrix $\mathbf{V}_C$ are the probabilities to move from vanishing states to the equivalence classes of vanishing states, and those of the $m \times k$ submatrix $\mathbf{V}_F$ are the probabilities to move from tangible states to the equivalence classes of tangible states. The elements of the $(l-k)\times(n-m)$ submatrix $\mathbf{W}_C$ are the probabilities to move from the equivalence classes of vanishing states to vanishing states, and those of the $k \times m$ submatrix $\mathbf{W}_F$ are the probabilities to move from the equivalence classes of tangible states to tangible states. We have

$$\mathbf{W}_r\mathbf{V}_r = \begin{pmatrix} \mathbf{W}_C\mathbf{V}_C & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_F\mathbf{V}_F \end{pmatrix} = \mathbf{I},$$

hence, $\mathbf{W}_C\mathbf{V}_C = \mathbf{I}$ and $\mathbf{W}_F\mathbf{V}_F = \mathbf{I}$.

Since tangible and vanishing states always belong to the equivalence classes of the same kind, the quotienting (by $\underline{\leftrightarrow}_{ss}$) and reordering (by moving vanishing states and their equivalence classes to the first positions) are permutable. The quotiented reordered TPM may only differ from the reordered quotiented TPM up to the order of the equivalence classes of tangible states and the order of the equivalence classes of vanishing states. To avoid such a difference, we rearrange the equivalence classes of the same kind in increasing order of the smallest indices of the states from them while keeping the equivalence classes of vanishing states at the first positions.

Then $\mathbf{P}_r\mathbf{V}_r = \mathbf{V}_r\mathbf{P}_{r\underline{\leftrightarrow}_{ss}}$ and $\mathbf{P}_{r\underline{\leftrightarrow}_{ss}} = \mathbf{W}_r\mathbf{P}_r\mathbf{V}_r$. We have

$$\mathbf{P}_r\mathbf{V}_r = \begin{pmatrix} \mathbf{C} & \mathbf{D} \\ \mathbf{E} & \mathbf{F} \end{pmatrix} \begin{pmatrix} \mathbf{V}_C & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_F \end{pmatrix} = \begin{pmatrix} \mathbf{C}\mathbf{V}_C & \mathbf{D}\mathbf{V}_F \\ \mathbf{E}\mathbf{V}_C & \mathbf{F}\mathbf{V}_F \end{pmatrix},$$

$$\mathbf{V}_r\mathbf{P}_{r\underline{\leftrightarrow}_{ss}} = \begin{pmatrix} \mathbf{V}_C & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_F \end{pmatrix} \begin{pmatrix} \mathbf{C}_{\underline{\leftrightarrow}_{ss}} & \mathbf{D}_{\underline{\leftrightarrow}_{ss}} \\ \mathbf{E}_{\underline{\leftrightarrow}_{ss}} & \mathbf{F}_{\underline{\leftrightarrow}_{ss}} \end{pmatrix} = \begin{pmatrix} \mathbf{V}_C\mathbf{C}_{\underline{\leftrightarrow}_{ss}} & \mathbf{V}_C\mathbf{D}_{\underline{\leftrightarrow}_{ss}} \\ \mathbf{V}_F\mathbf{E}_{\underline{\leftrightarrow}_{ss}} & \mathbf{V}_F\mathbf{F}_{\underline{\leftrightarrow}_{ss}} \end{pmatrix}.$$

Hence, $\mathbf{C}\mathbf{V}_C = \mathbf{V}_C\mathbf{C}_{\underline{\leftrightarrow}_{ss}}, \ \mathbf{D}\mathbf{V}_F = \mathbf{V}_C\mathbf{D}_{\underline{\leftrightarrow}_{ss}}, \ \mathbf{E}\mathbf{V}_C = \mathbf{V}_F\mathbf{E}_{\underline{\leftrightarrow}_{ss}}, \ \mathbf{F}\mathbf{V}_F = \mathbf{V}_F\mathbf{F}_{\underline{\leftrightarrow}_{ss}}$.

Let us show that $\mathbf{G}\mathbf{V}_C=\mathbf{V}_C\mathbf{G}_{\underline{\leftrightarrow}_{ss}}$. Since $\mathbf{G}=\sum_{k=0}^{\infty}\mathbf{C}^k$, it is sufficient to prove $\left(\sum_{k=0}^{l}\mathbf{C}^k\right)\mathbf{V}_C=\mathbf{V}_C\sum_{k=0}^{l}\mathbf{C}^k_{\underline{\leftrightarrow}_{ss}}$ by induction on $l\in\mathbb{N}$ and then take a limit $l\to\infty$.

- $l=0$
  We have $\left(\sum_{k=0}^{0}\mathbf{C}^k\right)\mathbf{V}_C=\mathbf{I}\mathbf{V}_C=\mathbf{V}_C=\mathbf{V}_C\mathbf{I}=\mathbf{V}_C\sum_{k=0}^{0}\mathbf{C}^k_{\underline{\leftrightarrow}_{ss}}$.
- $l\to l+1$
  Suppose that $\left(\sum_{k=0}^{l}\mathbf{C}^k\right)\mathbf{V}_C=\mathbf{V}_C\sum_{k=0}^{l}\mathbf{C}^k_{\underline{\leftrightarrow}_{ss}}$. Then we have $\left(\sum_{k=0}^{l+1}\mathbf{C}^k\right)\mathbf{V}_C=\left(\mathbf{I}+\mathbf{C}\sum_{k=0}^{l}\mathbf{C}^k\right)\mathbf{V}_C=\mathbf{V}_C+\mathbf{C}\mathbf{V}_C\sum_{k=0}^{l}\mathbf{C}^k_{\underline{\leftrightarrow}_{ss}}=$ $\mathbf{V}_C+\mathbf{V}_C\mathbf{C}_{\underline{\leftrightarrow}_{ss}}\sum_{k=0}^{l}\mathbf{C}^k_{\underline{\leftrightarrow}_{ss}}=\mathbf{V}_C\left(\mathbf{I}+\mathbf{C}_{\underline{\leftrightarrow}_{ss}}\sum_{k=0}^{l}\mathbf{C}^k_{\underline{\leftrightarrow}_{ss}}\right)=\mathbf{V}_C\sum_{k=0}^{l+1}\mathbf{C}^k_{\underline{\leftrightarrow}_{ss}}$.

Next, $\mathbf{P}^{\diamond}\mathbf{V}_F=(\mathbf{F}+\mathbf{E}\mathbf{G}\mathbf{D})\mathbf{V}_F=\mathbf{F}\mathbf{V}_F+\mathbf{E}\mathbf{G}\mathbf{D}\mathbf{V}_F=\mathbf{V}_F\mathbf{F}_{\underline{\leftrightarrow}_{ss}}+\mathbf{E}\mathbf{G}\mathbf{V}_C\mathbf{D}_{\underline{\leftrightarrow}_{ss}}=\mathbf{V}_F\mathbf{F}_{\underline{\leftrightarrow}_{ss}}+\mathbf{E}\mathbf{V}_C\mathbf{G}_{\underline{\leftrightarrow}_{ss}}\mathbf{D}_{\underline{\leftrightarrow}_{ss}}=\mathbf{V}_F\mathbf{F}_{\underline{\leftrightarrow}_{ss}}+\mathbf{V}_F\mathbf{E}_{\underline{\leftrightarrow}_{ss}}\mathbf{G}_{\underline{\leftrightarrow}_{ss}}\mathbf{D}_{\underline{\leftrightarrow}_{ss}}=\mathbf{V}_F(\mathbf{F}_{\underline{\leftrightarrow}_{ss}}+\mathbf{E}_{\underline{\leftrightarrow}_{ss}}\mathbf{G}_{\underline{\leftrightarrow}_{ss}}\mathbf{D}_{\underline{\leftrightarrow}_{ss}})=\mathbf{V}_F\mathbf{P}^{\diamond}_{\underline{\leftrightarrow}_{ss}}$. After left-multiplying by $\mathbf{W}_F$ the resulting equality $\mathbf{P}^{\diamond}\mathbf{V}_F=\mathbf{V}_F\mathbf{P}^{\diamond}_{\underline{\leftrightarrow}_{ss}}$, we finally get

$$(\mathbf{P}^{\diamond})_{\underline{\leftrightarrow}_{ss}}=\mathbf{W}_F\mathbf{P}^{\diamond}\mathbf{V}_F=\mathbf{P}^{\diamond}_{\underline{\leftrightarrow}_{ss}}.$$

□

A.3. **Proof of Proposition 10.** By Proposition 3, $(DR(G)\cup DR(G'))/_{\mathcal{R}}=((DR_T(G)\cup DR_T(G'))/_{\mathcal{R}})\uplus((DR_V(G)\cup DR_V(G'))/_{\mathcal{R}})$. Hence, $\forall\mathcal{H}\in(DR(G)\cup DR(G'))/_{\mathcal{R}}$, all states from $\mathcal{H}$ are tangible, when $\mathcal{H}\in(DR_T(G)\cup DR_T(G'))/_{\mathcal{R}}$, or all of them are vanishing, when $\mathcal{H}\in(DR_V(G)\cup DR_V(G'))/_{\mathcal{R}}$.

By definition of the steady-state PMFs for SMCs, $\forall s\in DR_V(G),\ \varphi(s)=0$ and $\forall s'\in DR_V(G'),\ \varphi'(s')=0$. Thus, $\forall\mathcal{H}\in(DR_V(G)\cup DR_V(G'))/_{\mathcal{R}},\ \sum_{s\in\mathcal{H}\cap DR(G)}\varphi(s)=\sum_{s\in\mathcal{H}\cap DR_V(G)}\varphi(s)=0=\sum_{s'\in\mathcal{H}\cap DR_V(G')}\varphi'(s')=\sum_{s'\in\mathcal{H}\cap DR(G')}\varphi'(s')$.

By Proposition 4 from [87], $\forall s\in DR_T(G),\ \varphi(s)=\frac{\psi(s)}{\sum_{\tilde{s}\in DR_T(G)}\psi(\tilde{s})}$ and $\forall s'\in DR_T(G'),\ \varphi'(s')=\frac{\psi'(s')}{\sum_{\tilde{s}'\in DR_T(G')}\psi'(\tilde{s}')}$, where ψ and ψ' are the steady-state PMFs for $DTMC(G)$ and $DTMC(G')$, respectively. Thus, $\forall\mathcal{H},\widetilde{\mathcal{H}}\in(DR_T(G)\cup DR_T(G'))/_{\mathcal{R}}$, $\sum_{s\in\mathcal{H}\cap DR(G)}\varphi(s)=\sum_{s\in\mathcal{H}\cap DR_T(G)}\varphi(s)=\sum_{s\in\mathcal{H}\cap DR_T(G)}\left(\frac{\psi(s)}{\sum_{\tilde{s}\in DR_T(G)}\psi(\tilde{s})}\right)=\frac{\sum_{s\in\mathcal{H}\cap DR_T(G)}\psi(s)}{\sum_{\tilde{s}\in DR_T(G)}\psi(\tilde{s})}=\frac{\sum_{s\in\mathcal{H}\cap DR_T(G)}\psi(s)}{\sum_{\widetilde{\mathcal{H}}}\sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR_T(G)}\psi(\tilde{s})}$ and $\sum_{s'\in\mathcal{H}\cap DR(G')}\varphi'(s')=\sum_{s'\in\mathcal{H}\cap DR_T(G')}\varphi'(s')=\sum_{s'\in\mathcal{H}\cap DR_T(G')}\left(\frac{\psi'(s')}{\sum_{\tilde{s}'\in DR_T(G')}\psi'(\tilde{s}')}\right)=\frac{\sum_{s'\in\mathcal{H}\cap DR_T(G')}\psi'(s')}{\sum_{\tilde{s}'\in DR_T(G')}\psi'(\tilde{s}')}=\frac{\sum_{s'\in\mathcal{H}\cap DR_T(G')}\psi'(s')}{\sum_{\widetilde{\mathcal{H}}}\sum_{\tilde{s}'\in\widetilde{\mathcal{H}}\cap DR_T(G')}\psi'(\tilde{s}')}$.

It remains to prove that $\forall\mathcal{H}\in(DR_T(G)\cup DR_T(G'))/_{\mathcal{R}},\ \sum_{s\in\mathcal{H}\cap DR_T(G)}\psi(s)=\sum_{s'\in\mathcal{H}\cap DR_T(G')}\psi'(s')$. Since $(DR(G)\cup DR(G'))/_{\mathcal{R}}=((DR_T(G)\cup DR_T(G'))/_{\mathcal{R}})\uplus((DR_V(G)\cup DR_V(G'))/_{\mathcal{R}})$, the previous equality is a consequence of the following: $\forall\mathcal{H}\in(DR(G)\cup DR(G'))/_{\mathcal{R}},\ \sum_{s\in\mathcal{H}\cap DR(G)}\psi(s)=\sum_{s'\in\mathcal{H}\cap DR(G')}\psi'(s')$.

Thus, we should prove that $\forall\mathcal{H}\in(DR(G)\cup DR(G'))/_{\mathcal{R}}\ \sum_{\{i\mid s_i\in\mathcal{H}\cap DR(G)\}}\psi_i=\sum_{\{j\mid s'_j\in\mathcal{H}\cap DR(G')\}}\psi'_j$.

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

$$\begin{cases}\psi\mathbf{P}=\psi\\ \psi\mathbf{1}^T=1\end{cases}.$$

Then, for all i $(1 \le i \le n)$, we have

$$\left\{ \begin{array}{l} \sum_{j=1}^{n} \mathcal{P}_{ji}\psi_j = \psi_i \\ \sum_{j=1}^{n} \psi_j = 1 \end{array} \right. .$$

By definition of $\mathcal{P}_{ij}$ $(1 \le i, j \le n)$ we have

$$\left\{ \begin{array}{l} \sum_{j=1}^{n} PM(s_j, s_i)\psi_j = \psi_i \\ \sum_{j=1}^{n} \psi_j = 1 \end{array} \right. .$$

Let $\mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ and $s_1, s_2 \in \mathcal{H}$. We have $\forall \widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ $\forall A \in \mathbb{N}_{fin}^{\mathcal{L}}$ $s_1 \xrightarrow{A}_{\mathcal{P}} \widetilde{\mathcal{H}} \Leftrightarrow s_2 \xrightarrow{A}_{\mathcal{P}} \widetilde{\mathcal{H}}$. Therefore, $PM(s_1, \widetilde{\mathcal{H}}) = \sum_{\{\Upsilon \mid \exists \tilde{s}_1 \in \widetilde{\mathcal{H}}\ s_1 \xrightarrow{\Upsilon} \tilde{s}_1\}} PT(\Upsilon, s_1) = \sum_{A \in \mathbb{N}_{fin}^{\mathcal{L}}} \sum_{\{\Upsilon \mid \exists \tilde{s}_1 \in \widetilde{\mathcal{H}}\ s_1 \xrightarrow{\Upsilon} \tilde{s}_1,\ \mathcal{L}(\Upsilon) = A\}} PT(\Upsilon, s_1) = \sum_{A \in \mathbb{N}_{fin}^{\mathcal{L}}} PM_A(s_1, \widetilde{\mathcal{H}}) = \sum_{A \in \mathbb{N}_{fin}^{\mathcal{L}}} PM_A(s_2, \widetilde{\mathcal{H}}) = \sum_{A \in \mathbb{N}_{fin}^{\mathcal{L}}} \sum_{\{\Upsilon \mid \exists \tilde{s}_2 \in \widetilde{\mathcal{H}}\ s_2 \xrightarrow{\Upsilon} \tilde{s}_2,\ \mathcal{L}(\Upsilon) = A\}} PT(\Upsilon, s_2) = \sum_{\{\Upsilon \mid \exists \tilde{s}_2 \in \widetilde{\mathcal{H}}\ s_2 \xrightarrow{\Upsilon} \tilde{s}_2\}} PT(\Upsilon, s_2) = PM(s_2, \widetilde{\mathcal{H}})$. Since we have the previous equality for all $s_1, s_2 \in \mathcal{H}$, we can denote $PM(\mathcal{H}, \widetilde{\mathcal{H}}) = PM(s_1, \widetilde{\mathcal{H}}) = PM(s_2, \widetilde{\mathcal{H}})$. Note that transitions from the states of $DR(G)$ always lead to those from the same set, hence, $\forall s \in DR(G)$ $PM(s, \widetilde{\mathcal{H}}) = PM(s, \widetilde{\mathcal{H}} \cap DR(G))$. The same is true for $DR(G')$.

Let $\mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$. We sum the left and right parts of the first equation from the system above for all i such that $s_i \in \mathcal{H} \cap DR(G)$. The result is

$$\sum_{\{i \mid s_i \in \mathcal{H} \cap DR(G)\}} \sum_{j=1}^{n} PM(s_j, s_i)\psi_j = \sum_{\{i \mid s_i \in \mathcal{H} \cap DR(G)\}} \psi_i.$$

Let us denote the aggregate steady-state PMF for $DTMC(G)$ by $\psi_{\mathcal{H} \cap DR(G)} = \sum_{\{i \mid s_i \in \mathcal{H} \cap DR(G)\}} \psi_i$. Then, for the left part of the equation above, we get $\sum_{\{i \mid s_i \in \mathcal{H} \cap DR(G)\}} \sum_{j=1}^{n} PM(s_j, s_i)\psi_j = \sum_{j=1}^{n} \psi_j \sum_{\{i \mid s_i \in \mathcal{H} \cap DR(G)\}} PM(s_j, s_i) = \sum_{j=1}^{n} PM(s_j, \mathcal{H})\psi_j = \sum_{\widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}} \sum_{\{j \mid s_j \in \widetilde{\mathcal{H}} \cap DR(G)\}} PM(s_j, \mathcal{H})\psi_j = \sum_{\widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}} \sum_{\{j \mid s_j \in \widetilde{\mathcal{H}} \cap DR(G)\}} PM(\widetilde{\mathcal{H}}, \mathcal{H})\psi_j = \sum_{\widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}} PM(\widetilde{\mathcal{H}}, \mathcal{H}) \sum_{\{j \mid s_j \in \widetilde{\mathcal{H}} \cap DR(G)\}} \psi_j = \sum_{\widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}} PM(\widetilde{\mathcal{H}}, \mathcal{H})\psi_{\widetilde{\mathcal{H}} \cap DR(G)}$.

For the left part of the second equation from the system above, we get $\sum_{j=1}^{n} \psi_j = \sum_{\widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}} \sum_{\{j \mid s_j \in \widetilde{\mathcal{H}} \cap DR(G)\}} \psi_j = \sum_{\widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}} \psi_{\widetilde{\mathcal{H}} \cap DR(G)}$.

Thus, the aggregate linear equation system for $DTMC(G)$ is

$$\left\{ \begin{array}{l} \sum_{\widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}} PM(\widetilde{\mathcal{H}}, \mathcal{H})\psi_{\widetilde{\mathcal{H}} \cap DR(G)} = \psi_{\mathcal{H} \cap DR(G)} \\ \sum_{\widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}} \psi_{\widetilde{\mathcal{H}} \cap DR(G)} = 1 \end{array} \right. .$$

Let us denote the aggregate steady-state PMFs for $DTMC(G')$ by $\psi'_{\mathcal{H} \cap DR(G')} = \sum_{\{j \mid s'_j \in \mathcal{H} \cap DR(G')\}} \psi'_j$. Then the aggregate linear equation system for $DTMC(G')$ is

$$\left\{ \begin{array}{l} \sum_{\widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}} PM(\widetilde{\mathcal{H}}, \mathcal{H})\psi'_{\widetilde{\mathcal{H}} \cap DR(G')} = \psi'_{\mathcal{H} \cap DR(G')} \\ \sum_{\widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}} \psi'_{\widetilde{\mathcal{H}} \cap DR(G')} = 1 \end{array} \right. .$$

Let $(DR(G) \cup DR(G'))/_{\mathcal{R}} = \{\mathcal{H}_1, \ldots, \mathcal{H}_l\}$. Then the aggregate steady-state PMFs $\psi_{\mathcal{H}_k \cap DR(G)}$ and $\psi'_{\mathcal{H}_k \cap DR(G')}$ $(1 \le k \le l)$ satisfy the same aggregate system of $l+1$ linear equations with l independent equations and l unknowns. The aggregate linear equation system has a unique solution, when a single aggregate steady-state PMF exists. This is the case here, since in [87] we have demonstrated that $DTMC(G)$ has a single steady state iff $SMC(G)$ has, and aggregation preserves this property [30]. Hence, $\psi_{\mathcal{H}_k \cap DR(G)} = \psi'_{\mathcal{H}_k \cap DR(G')}$ $(1 \leq k \leq l)$. $\square$

References

[1] W.M.P. van der Aalst, K.M. van Hee, H.A. Reijers, *Analysis of discrete-time stochastic Petri nets*, Statistica Neerlandica, **54**:2 (2000), 237–255. http://tmitwww.tm.tue.nl/staff/hreijers/H.A. Reijers Bestanden/Statistica.pdf. MR1794979

[2] A. Aldini, M. Bernardo, F. Corradini, *A process algebraic approach to software architecture design*, Springer, 2010. Zbl 1255.68004

[3] C. Autant, Ph. Schnoebelen, *Place bisimulations in Petri nets*, Lecture Notes in Computer Science, **616** (1992), 45–61.

[4] G. Balbo, *Introduction to stochastic Petri nets*, Lecture Notes in Computer Science, **2090** (2001), 84–155. Zbl 0990.68092

[5] G. Balbo, *Introduction to generalized stochastic Petri nets*, Lecture Notes in Computer Science, **4486** (2007), 83–131. Zbl 1323.68400

[6] E. Bartocci, P. Lió, *Computational modeling, formal analysis, and tools for systems biology*, PLoS Computational Biology **12**:1 (2016), e1004591.

[7] F. Bause, P.S. Kritzinger, *Stochastic Petri nets: an introduction to the theory*, Vieweg Verlag, 2002. http://ls4-www.cs.tu-dortmund.de/cms/de/home/bause/bause_kritzinger_spn_book_print.pdf Zbl 1013.60065

[8] J.A. Bergstra, J.W. Klop, *Algebra of communicating processes with abstraction*, Theoretical Computer Science, **37** (1985), 77–121. MR0796314

[9] M. Bernardo, *Theory and application of extended Markovian process algebra*, Ph.D. thesis, 276 p., University of Bologna, Italy, 1999. http://www.sti.uniurb.it/bernardo/documents/phdthesis.pdf

[10] M. Bernardo, *A survey of Markovian behavioral equivalences*, Lecture Notes in Computer Science, **4486** (2007), 180–219. Zbl 1323.68402

[11] M. Bernardo, *Non-bisimulation-based Markovian behavioral equivalences*, Journal of Logic and Algebraic Programming, **72** (2007), 3–49. Zbl 1121.68077

[12] M. Bernardo, *On the tradeoff between compositionality and exactness in weak bisimilarity for integrated-time Markovian process calculi*, Theoretical Computer Science, **563** (2015), 99–143. MR3286633

[13] M. Bernardo, *ULTraS at work: compositionality metaresults for bisimulation and trace semantics*, Journal of Logical and Algebraic Methods in Programming, **94** (2018), 150–182. Zbl 1381.68195

[14] M. Bernardo, S. Botta, *Modal logic characterization of Markovian testing and trace equivalences*, Electronic Notes in Theoretical Computer Science, **169** (2006), 7–18. Zbl 1276.68119

[15] M. Bernardo, S. Botta, *A survey of modal logics characterizing behavioural equivalences for non-deterministic and stochastic systems*, Mathematical Structures in Computer Science, **18** (2008), 29–55. MR2459612

[16] M. Bernardo, M. Bravetti, *Reward based congruences: can we aggregate more?* Lecture Notes in Computer Science, **2165** (2001), 136–151. MR1904353

[17] M. Bernardo, L. Donatiello, R. Gorrieri, *A formal approach to the integration of performance aspects in the modeling and analysis of concurrent systems*, Information and Computation, **144**:2 (1998), 83–154. MR1632840

[18] M. Bernardo, R. Gorrieri, *A tutorial on EMPA: a theory of concurrent processes with nondeterminism, priorities, probabilities and time*, Theoretical Computer Science, **202** (1998), 1–54. MR1626813

[19] M. Bernardo, R. De Nicola, M. Loreti, *A uniform framework for modeling nondeterministic, probabilistic, stochastic, or mixed processes and their behavioral equivalences*, Information and Computation, **225** (2013), 29–82. Zbl 1358.68210

[20] M. Bernardo, L. Tesei, *Encoding timed models as uniform labeled transition systems*, Lecture Notes in Computer Science, **8168** (2013), 104–118.

[21] E. Best, R. Devillers, J.G. Hall, *The box calculus: a new causal algebra with multi-label communication*, Lecture Notes in Computer Science, **609** (1992), 21–69. MR1253529

[22] E. Best, R. Devillers, M. Koutny, *Petri net algebra*, EATCS Monographs on Theoretical Computer Science, Springer, 2001. MR1932732

[23] E. Best, M. Koutny, *A refined view of the box algebra*, Lecture Notes in Computer Science, **935** (1995), 1–20. MR1461021

[24] T. Bolognesi, F. Lucidi, S. Trigila, *From timed Petri nets to timed LOTOS*, Proc. IFIP WG 6.1 10^{th} Int. Symposium on Protocol Specification, Testing and Verification 1990, Ottawa, Canada, 1–14, North-Holland, Amsterdam, The Netherlands, 1990.

[25] N. Bonzanni, K.A. Feenstra, W. Fokkink, E. Krepska, *What can formal methods bring to systems biology?* Lecture Notes in Computer Science, **5850** (2009), 16–22.

[26] A.A. Borovkov, *Probability theory*, Universitext (UTX) series, Springer, 2013. Zbl 1297.60002

[27] M. Bravetti, *Specification and analysis of stochastic real-time systems*, Ph.D. thesis, 432 p., University of Bologna, Italy, 2002. http://www.cs.unibo.it/~bravetti/papers/phdthesis.ps.gz

[28] M. Bravetti, M. Bernardo, R. Gorrieri, *Towards performance evaluation with general distributions in process algebras*, Lecture Notes in Computer Science, **1466** (1998), 405–422. MR1683349

[29] P. Buchholz, *Markovian process algebra: composition and equivalence*, Proc. 2^{nd} Int. Workshop on Process Algebras and Performance Modelling (PAPM) 1994 (U. Herzog, M. Rettelbach, eds.), Regensberg / Erlangen, Germany, July 1994, Arbeitsberichte des IMMD, **27**:4 (1994), 11–30.

[30] P. Buchholz, *Exact and ordinary lumpability in finite Markov chains*, Journal of Applied Probability, **31**:1 (1994), 59–75.

[31] P. Buchholz, *A notion of equivalence for stochastic Petri nets*, Lecture Notes in Computer Science, **935** (1995), 161–180. MR1461026

[32] P. Buchholz, *Iterative decomposition and aggregation of labeled GSPNs*, Lecture Notes in Computer Science, **1420** (1998), 226–245.

[33] P. Buchholz, I.V. Tarasyuk, *Net and algebraic approaches to probabilistic modeling*, Joint Novosibirsk Computing Center and Institute of Informatics Systems Bulletin, Series Computer Science, **15** (2001), 31–64. Zbl 1004.68112

[34] P. Buchholz, I.V. Tarasyuk, *Equivalences for stochastic Petri nets and stochastic process algebras*, Vestnik, Quartal Journal of Novosibirsk State University, Series: Mathematics, Mechanics and Informatics, **6**:1 (2006), 14–42 (in Russian). Zbl 1249.68126

[35] I. Christoff, *Testing equivalence and fully abstract models of probabilistic processes*, Lecture Notes in Computer Science, **458** (1990), 126–140. MR1082160

[36] N. Coste, H. Hermanns, E. Lantreibecq, W. Serwe, *Towards performance prediction of compositional models in industrial GALS designs*, Lecture Notes in Computer Science, **5643** (2009), 204–218. Zbl 1242.68005

[37] V. Danos, J. Feret, W. Fontana, R. Harmer, J. Krivine, *Rule-based modelling of cellular signalling*, Lecture Notes in Computer Science, **4703** (2007), 17–41. Zbl 1151.68723

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

[39] Ch. Eisentraut, H. Hermanns, J. Schuster, A. Turrini, L. Zhang, *The quest for minimal quotients for probabilistic automata*, Lecture Notes in Computer Science, **7795** (2013), 16–31. Zbl 1381.68115

[40] D. Gilbert, M. Heiner, S. Lehrack, *A unifying framework for modelling and analysing biochemical pathways using Petri nets*, Lecture Notes in Computer Science, **4695** (2007), 200–216.

[41] R.J. van Glabbeek, S.A. Smolka, B. Steffen, *Reactive, generative, and stratified models of probabilistic processes*, Information and Computation, **121**:1 (1995), 59–80. MR1347332

[42] M.C. Guenther, N.J. Dingle, J.T. Bradley, W.J. Knottenbelt, *Passage-time computation and aggregation strategies for large semi-Markov processes*, Performance Evaluation, **68** (2011), 221–236.

[43] H.M. Hanish, *Analysis of place/transition nets with timed-arcs and its application to batch process control*, Lecture Notes in Computer Science, **691** (1993), 282–299.

[44] H. Hermanns, *Interactive Markov chains: the quest for quantified quality*, Lecture Notes in Computer Science, **2428** (2002). Zbl 1012.68142

[45] H. Hermanns, M. Rettelbach, *Syntax, semantics, equivalences and axioms for MTIPP*, Proc. 2^{nd} Int. Workshop on Process Algebras and Performance Modelling (PAPM) 1994 (U. Herzog, M. Rettelbach, eds.), Regensberg / Erlangen, Germany, July 1994, Arbeitsberichte

des IMMD, **27**:4 (1994), 71–88. http://ftp.informatik.uni-erlangen.de/local/inf7/papers/Hermanns/syntax_semantics_equivalences_axioms_for_MTIPP.ps.gz

[46] J. Hillston, *A compositional approach to performance modelling*, Cambridge University Press, Cambridge, UK, 1996. http://www.dcs.ed.ac.uk/pepa/book.pdf MR1427945

[47] C.A.R. Hoare, *Communicating sequential processes*, Prentice-Hall, London, UK, 1985. http://www.usingcsp.com/cspbook.pdf MR0805324

[48] A. Horváth, M., Paolieri, L. Ridi, E. Vicario, *Probabilistic model checking of non-Markovian models with concurrent generally distributed timers*, Proc. 8^{th} Int. Conf. on Quantitative Evaluation of Systems (QEST) 2011, Aachen, Germany, September 2011, 131–140, IEEE Computer Society Press, 2011.

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

[50] C.-C. Jou, S.A. Smolka, *Equivalences, congruences and complete axiomatizations for probabilistic processes*, Lecture Notes in Computer Science, **458** (1990), 367–383. MR1082173

[51] M. Koutny, *A compositional model of time Petri nets*, Lecture Notes in Computer Science, **1825** (2000), 303–322.

[52] V.G. Kulkarni, *Modeling and analysis of stochastic systems*, Texts in Statistical Science, **84**, Chapman and Hall / CRC Press, 2010. Zbl 1191.60003

[53] L. Lakatos, L. Szeidl, M. Telek, *Introduction to queueing systems with telecommunication applications*, Springer Nature, Cham, Switzerland, 2019. Zbl 1415.60001

[54] K.G. Larsen, A. Skou, *Bisimulation through probabilistic testing*, Information and Computation, **94**:1 (1991), 1–28. MR1123153

[55] D. Latella, M. Massink, E.P. de Vink, *Bisimulation of labelled state-to-function transition systems coalgebraically*, Logical Methods in Computer Science, **11**:4:16 (2015), 1–40. Zbl 1448.68345

[56] H. Macià, V. Valero, D.C. Cazorla, F. Cuartero, *Introducing the iteration in sPBC*, Lecture Notes in Computer Science, **3235** (2004), 292–308. Zbl 1110.68420

[57] H. Macià, V. Valero, F. Cuartero, D. de Frutos, *A congruence relation for sPBC*, Formal Methods in System Design, **32**:2 (2008), 85–128. Zbl 1138.68040

[58] H. Macià, V. Valero, F. Cuartero, M.C. Ruiz, *sPBC: a Markovian extension of Petri box calculus with immediate multiactions*, Fundamenta Informaticae, **87**:3–4 (2008), 367–406. Zbl 1154.68092

[59] H. Macià, V. Valero, F. Cuartero, M.C. Ruiz, I.V. Tarasyuk, *Modelling a video conference system with sPBC*, Applied Mathematics and Information Sciences **10**:2 (2016), 475–493.

[60] H. Macià, V. Valero, D. de Frutos, *sPBC: a Markovian extension of finite Petri box calculus*, Proc. 9^{th} IEEE Int. Workshop on Petri Nets and Performance Models (PNPM) 2001, Aachen, Germany, 207–216, IEEE Computer Society Press, 2001. http://www.info-ab.uclm.es/retics/publications/2001/pnpm01.ps

[61] A. Marin, C. Piazza, S. Rossi, *Proportional lumpability*, Lecture Notes in Computer Science, **11750** (2019), 265–281. Zbl 1434.68055

[62] A. Marin, C. Piazza, S. Rossi, *Proportional lumpability and proportional bisimilarity*, Acta Informatica, **59**:2–3 (2022), 211–244. Zbl 07528110

[63] Marin A., Piazza C., Rossi S. *Proportional lumpability and proportional bisimilarity. Acta Informatica* **59(2–3)**, p. , June 2022.

[64] O. Marroquín, D. de Frutos, *TPBC: timed Petri box calculus*, Technical Report, Departamento de Sistemas Infofmáticos y Programación, Universidad Complutense de Madrid, Spain, 2000 (in Spanish).

[65] O. Marroquín, D. de Frutos, *Extending the Petri box calculus with time*, Lecture Notes in Computer Science, **2075** (2001), 303–322. Zbl 0986.68082

[66] M.A. Marsan, *Stochastic Petri nets: an elementary introduction*, Lecture Notes in Computer Science, **424** (1990), 1–29.

[67] M.A. Marsan, G. Balbo, G. Conte, S. Donatelli, G. Franceschinis, *Modelling with generalised stochastic Petri nets*, Wiley Series in Parallel Computing, John Wiley and Sons, 1995. http://www.di.unito.it/~greatspn/GSPN-Wiley/ Zbl 0843.68080

[68] Ph.M. Merlin, D.J. Farber, *Recoverability of communication protocols: implications of a theoretical study*, IEEE Transactions on Communications, **24**:9 (1976), 1036–1043. Zbl 0362.68096

[69] R.A.J. Milner, *Communication and concurrency*, Prentice-Hall, Upper Saddle River, NJ, USA, 1989. Zbl 0683.68008

[70] M.K. Molloy, *On the integration of the throughput and delay measures in distributed processing models*, Ph.D. thesis, Report, **CSD-810-921**, 108 p., University of California, Los Angeles, CA, USA, 1981.

[71] M.K. Molloy, *Discrete time stochastic Petri nets*, IEEE Transactions on Software Engineering, **11**:4 (1985), 417–423. MR0788999

[72] A. Niaouris, *An algebra of Petri nets with arc-based time restrictions*, Lecture Notes in Computer Science, **3407** (2005), 447–462. Zbl 1109.68076

[73] A. Niaouris, M. Koutny, *An algebra of timed-arc Petri nets*, Technical Report, **CS-TR-895**, 60 p., School of Computer Science, University of Newcastle upon Tyne, UK, 2005. http://www.cs.ncl.ac.uk/publications/trs/papers/895.pdf

[74] R. Paige, R.E. Tarjan, *Three partition refinement algorithms*, SIAM Journal of Computing, **16**:6 (1987), 973–989.

[75] C. Priami, *Language-based performance prediction for distributed and mobile systems*, Information and Computation, **175**:2 (2002), 119–145. MR1911524

[76] C. Ramchandani, *Performance evaluation of asynchronous concurrent systems by timed Petri nets*, Ph.D. thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1973.

[77] S.M. Ross, *Stochastic processes*, John Wiley and Sons, New York, USA, 1996. MR1373653

[78] S.M. Ross, *Introduction to probability models*, Academic Press, Elsevier, UK, 2019. Zbl 1408.60002

[79] A.N. Shiryaev, *Probability-2*, Graduate Texts in Mathematics (GTM) series, Springer, New York, 2019.

[80] W.J. Stewart, *Probability, Markov chains, queues, and simulation. The mathematical basis of performance modeling*, Princeton University Press, Princeton, NJ, USA, 2009. Zbl 1176.60003

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

[82] I.V. Tarasyuk, *Iteration in discrete time stochastic Petri box calculus*, Bulletin of the Novosibirsk Computing Center, Series Computer Science, IIS Special Issue, **24** (2006), 129–148. Zbl 1249.68132

[83] I.V. Tarasyuk, *Stochastic Petri box calculus with discrete time*, Fundamenta Informaticae, **76**:1–2 (2007), 189–218. MR2293057

[84] I.V. Tarasyuk, *Equivalences for behavioural analysis of concurrent and distributed computing systems*, Geo Academic Publisher, Novosibirsk, Russia, 2007 (in Russian).

[85] I.V. Tarasyuk, *Equivalence relations for modular performance evaluation in dtsPBC*, Mathematical Structures in Computer Science, **24**:1 (2014), e240103. MR3183269

[86] I.V. Tarasyuk, *Discrete time stochastic and deterministic Petri box calculus dtsdPBC*, Siberian Electronic Mathematical Reports, **17** (2020), 1598–1679. Zbl 1448.68352

[87] I.V. Tarasyuk, *Performance evaluation in stochastic process algebra dtsdPBC*, Siberian Electronic Mathematical Reports, **18**:2 (2021), 1105–1145. Zbl 1482.68156

[88] I.V. Tarasyuk, *Stochastic bisimulation and performance evaluation in discrete time stochastic and deterministic Petri box calculus dtsdPBC*, **hal-02573419v4**, 142 p., HAL Open Archives, France, January 2023. https://hal.science/hal-02573419v4/file/dtsdpbchal3rv.pdf.

[89] I.V. Tarasyuk, H. Macià, V. Valero, *Discrete time stochastic Petri box calculus with immediate multiactions*, Technical Report, **DIAB-10-03-1**, 25 p., Department of Computer Systems, High School of Computer Science Engineering, University of Castilla - La Mancha, Albacete, Spain, 2010. http://www.dsi.uclm.es/descargas/technicalreports/DIAB-10-03-1/dtsipbc.pdf

[90] I.V. Tarasyuk, H. Macià, V. Valero, *Discrete time stochastic Petri box calculus with immediate multiactions dtsiPBC*, 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 (J. Bradley, K. Heljanko, W. Knottenbelt, N. Thomas, eds.), London, UK, 2012, Electronic Notes in Theoretical Computer Science, **296** (2013), 229–252.

[91] I.V. Tarasyuk, H. Macià, V. Valero, *Performance analysis of concurrent systems in algebra dtsiPBC*, Programming and Computer Software, **40**:5 (2014), 229–249. Zbl 1339.68033

[92] I.V. Tarasyuk, H. Macià, V. Valero, *Stochastic process reduction for performance evaluation in dtsiPBC*, Siberian Electronic Mathematical Reports, **12** (2015), 513–551. MR3493774

[93] I.V. Tarasyuk, H. Macià, V. Valero, *Stochastic equivalence for performance analysis of concurrent systems in dtsiPBC*, Siberian Electronic Mathematical Reports, **15** (2018), 1743–1812. Zbl 1414.60062

[94] M. Timmer, J.-P. Katoen, J. van de Pol, M.I.A. Stoelinga, *Efficient modelling and generation of Markov automata*, Lecture Notes in Computer Science, **7454** (2012), 364–379. Zbl 1364.68295

[95] K.S. Trivedi, *Probability and statistics with reliability, queuing, and computer science applications*, John Wiley and Sons, 2016. Zbl 1344.60003

[96] V. Valero, M.E. Cambronero, *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, **23**:6 (2017), 570–594.

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

[98] R. Zijal, *Discrete time deterministic and stochastic Petri nets*, Proc. Int. Workshop on Quality of Communication-Based Systems 1994, Technical University of Berlin, Germany, 123–136, Kluwer Academic Publishers, 1995. Zbl 0817.68111

[99] R. Zijal, *Analysis of discrete time deterministic and stochastic Petri nets*, Ph.D. thesis, Technical University of Berlin, Germany, 1997.

[100] R. Zijal, R. German, *A new approach to discrete time stochastic Petri nets*, Proc. 11^{th} Int. Conf. on Analysis and Optimization of Systems, Discrete Event Systems (DES) 1994 (G. Cohen, J.-P. Quadrat, eds.), Sophia-Antipolis, France, 1994, Lecture Notes in Control and Information Sciences, **199** (1994), 198–204.

Igor Valerievich Tarasyuk
A.P. Ershov Institute of Informatics Systems,
Siberian Branch of the Russian Academy of Sciences,
Acad. Lavrentiev pr. 6,
630090 Novosibirsk, Russian Federation
Email address: itar@iis.nsk.su