Equivalence relations for behaviour-preserving reduction and modular performance evaluation in *dtsPBC* *

IGOR V. TARASYUK

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

Abstract

In the last decades, a number of stochastic enrichments of process algebras was constructed to specify stochastic processes within the well-developed framework of algebraic calculi. In 2003, a continuous time stochastic extension *sPBC* of finite Petri box calculus (*PBC*) was enriched with the iteration operator by H. MACIÀ, V. VALERO, D. CAZORLA and F. CUARTERO. In 2006, the author added iteration to the discrete time stochastic extension *dtsPBC* of finite *PBC*. In this paper, in the framework of the *dtsPBC* with iteration, we define a variety of stochastic equivalences. They allow one to identify stochastic processes with similar behaviour that are however differentiated by the operational and denotational semantics of the calculus. The interrelations of all the introduced equivalences are investigated. It is explained how the equivalences we propose can be used to reduce transition systems of expressions. A logical characterization of the equivalences is presented via formulas of the new probabilistic modal logics. We demonstrate how to apply the equivalences to compare the stationary behaviour. A problem of preservation of the equivalences by algebraic operations is discussed. As a result, we define an equivalence that is a congruence relation. Finally, two case studies of performance evaluation in the algebra are presented.

Keywords: stochastic Petri net, stochastic process algebra, Petri box calculus, iteration, discrete time, transition system, operational semantics, dts-box, denotational semantics, empty loop, stochastic equivalence, reduction, modal logic, stationary behaviour, congruence relation, performance evaluation.

Contents

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

1 Introduction

Stochastic Petri nets (SPNs) are a well-known model for quantitative analysis of discrete dynamic event systems proposed initially in [98]. Essentially, SPNs are a high level language for specification and performance analysis of concurrent systems. A stochastic process corresponding to this formal model is a Markov chain generated and analyzed by well-developed algorithms and methods. Firing probabilities distributed along continuous or discrete time scale are associated with transitions of an SPN. Thus, there exist SPNs with continuous [55, 98] and discrete [99] time. Markov chains of the corresponding types are associated with the SPNs. As a rule, for SPNs with continuous time (CTSPNs), exponential or phase distributions of transition probabilities are used. For SPNs with discrete time (DTSPNs), geometric or combinations of geometric distributions are usually used. Transitions of CTSPNs fire one by one at continuous time moments. Hence, the semantics of this model is an interleaving one. In this semantics, parallel computations are modeled by all possible execution sequences of their components. Transitions of DTSPNs fire concurrently in steps at discrete time moments. Hence, this model has a step semantics. In this semantics, parallel computations are modeled by sequences of concurrent occurrences (steps) of their components. In [41, 42], a labeling for transitions of CTSPNs with action names was proposed. The labeling allows SPNs to model processes with functionally similar components: the transitions corresponding to the similar components are labeled by the same action. Moreover, one can compare labeled SPNs by different behavioural equivalences, and this makes possible to check stochastic processes specified by labeled SPNs for functional similarity. Therefore, one can compare both functional and performance properties, and labeled SPNs turn into a formalism for qualitative and quantitative analysis.

Algebraic calculi occupy a special place among formal models for specification of concurrent systems and analysis of their behavioral properties. In such process algebras (PAs), a system or a process is specified by an algebraic formula. Verification of the properties is accomplished at a syntactic level by means of well-developed systems of equivalences, axioms and inference rules. The best-known of the first PAs are Theory of Communicating Sequential Processes ($TCSP$) [67] and Calculus of Communicating Systems (CCS) [96]. Process algebras have been acknowledged to be very suitable formalism to operate with real time and stochastic systems as well. In the last years, stochastic extensions of PAs, called stochastic process algebras (SPAs), became very popular as a modeling framework. SPAs do not just specify actions which can occur (qualitative features) as usual process algebras, but they associate some quantitative parameters with actions (quantitative characteristics). The most popular SPAs proposed so far are Markovian Timed Processes for Performance Evaluation ($MTIPP$) [70], Performance Evaluation Process Algebra ($PEPA$) [64] and Extended Markovian Process Algebra ($EMPA$) [23].

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

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

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

An extension of CCS with probabilities and time, called $TPCCS$, was defined in [60]. An enrichment of Basic Process Algebra (BPA) with probabilistic choice, $prBPA$, as well as extension of the latter with the parallel composition operator named ACP_π^+ have been proposed in [2]. A stochastic process calculus Priced Process Algebra (PPA) based on CCS was constructed in [117,120]. The papers [27,39,51,127] propose a variety of other SPAs. A standard way for probabilistic extension of process algebras into the calculi of probabilistic transition systems was described in [73].

Process algebras allow one to specify processes in a compositional way via an expressive formal syntax. On the other hand, Petri nets provide one with an ability for visual representation of a process structure and execution. Hence, the relationship between SPNs and SPAs is of particular interest. To combine advantages of both models, a semantics of algebraic formulas in terms of Petri nets is usually defined. In the stochastic case, the Markov chain of the stochastic process specified by an SPA formula is built based on the state transition graph of the corresponding SPN.

Petri box calculus (PBC) is a flexible and expressive process algebra based on calculi CCS [96]. Note that some operations of PBC are similar to those of the Algebra of Finite Processes (AFP_0) [81]. PBC was proposed fifteen years ago [12], and it was well explored since that time [10, 13–18, 29, 30, 32, 46–48, 52, 66, 76, 79, 82]. Its goal was to propose a compositional semantics for high level constructs of concurrent programming languages in terms of elementary Petri nets. Thus, PBC serves as a bridge between theory and applications. Formulas of PBC are combined not from single actions (including the invisible one) and variables, like in CCS, but from multisets of elementary actions and their conjugates, called multiactions (*basic formulas*) as well. The empty multiset of actions is allowed that is considered as the silent multiaction specifying some invisible or internal activity. In contrast to CCS, concurrency and synchronization are different operations (*concurrent constructs*). Synchronization is defined as a unary multi-way stepwise operation based on communication of actions and their conjugates. The CCS approach with conjugate matching labels was extended to define synchronization in PBC. This approach was preferred as being more flexible and compositional than that of the process algebras $TCSP$ and $COSY$ [22] where synchronization is accomplished over common action names. Moreover, the synchronization operation of PBC is asynchronous in contrast to the approach of Synchronous CCS ($SCCS$) [96] where it is synchronous. The other fundamental operations are sequence and choice (*sequential constructs*). The calculus includes also restriction and relabeling (*abstraction constructs*). To specify infinite processes, refinement, recursion and iteration operations were added (*hierarchical constructs*). Thus, unlike CCS, PBC has an additional iteration construction to specify infinite behaviour in the cases when finite Petri nets can be used as the semantic interpretation. For PBC, a denotational semantics was proposed in terms of a subclass of Petri nets equipped with an interface and considered up to isomorphism. This subclass is called Petri boxes. The calculus PBC has a step operational semantics in terms of labeled transition systems, based on the rules of structural operational semantics (SOS) [122]. 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). Note that we do not reason in terms of a big-step (natural) [75] or small-step (structural) [122] operational semantics here, and that PBC (and all its extensions to be mentioned further) have a small-step operational semantics, in that terminology. Pomset operational semantics of PBC was defined in [82] such that the partial order information was extracted from "decorated" step traces. In these step sequences, multiactions were annotated with an information on the relative position of the expression part they were derived from. More detailed comparison of PBC with other well-known process algebras and the reasoning about importance of non-interleaving semantics can be found in [12,15]. In the last years, several extensions of PBC were presented.

To specify systems with time constraints, such as real time systems, deterministic (fixed) or nondeterministic (interval) time delays are used. A time extension of PBC with a nondeterministic time model, called time Petri box calculus ($tPBC$), was proposed in [83]. In $tPBC$, timing information is added by associating time intervals (the earliest and the latest firing time) with instantaneous *actions*. Its denotational semantics was defined in terms of a subclass of labeled time Petri nets (LtPNs), based on tPNs [95] and called time Petri boxes (ct-boxes). $tPBC$ has a step time operational semantics in terms of labeled transition systems. Another time enrichment of PBC, called Timed Petri box calculus ($TPBC$), was defined in [93, 94], it accommodates a deterministic model of time. In contrast to $tPBC$, multiactions of $TPBC$ are not instantaneous, but have time durations. Additionally, in $TPBC$ there exist no "illegal" multiaction occurrences, unlike $tPBC$. The complexity of "illegal" occurrences mechanism was one of the main intentions to construct $TPBC$ though this calculus appeared to be more complicated than $tPBC$. The denotational semantics of $TPBC$ was defined in terms of a subclass of labeled Timed Petri nets (LTPNs), based on TPNs [126] and called Timed Petri boxes (T-boxes). $TPBC$ has a step timed operational semantics in terms of labeled transition systems. Note that $tPBC$ and $TPBC$ differ in ways they capture time information, and they are not in competition but complement each other. The third time extension of PBC, called arc time Petri box calculus ($atPBC$), was constructed in [118,119], and it implements a nondeterministic time. In $atPBC$, multiactions are associated with time delay intervals. Its denotational semantics was defined on a subclass of labeled arc time Petri nets (atPNs), where time restrictions are associated with the arcs, called arc time Petri boxes (at-boxes). $atPBC$ possesses a step time operational semantics in terms of labeled transition systems. Further, all the calculi $tPBC$, $TPBC$ and $atPBC$ apply the discrete time approach, but only $tPBC$ and $atPBC$ have immediate (multi)actions.

The set of states for the systems with deterministic or nondeterministic delays often differs drastically from that for the timeless systems, hence, the analysis results for untimed systems may be not valid for the time ones. To solve this problem, stochastic delays are considered, which are the random variables with a (discrete or continuous) probability distribution. If the random variables governing delays have an infinite support then the corresponding SPA can exhibit all the same behaviour as its underlying untimed PA. A stochastic extension of PBC, called stochastic Petri box calculus ($sPBC$), was proposed in [90, 103, 108–110, 112–114]. 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. Just a finite part of PBC was used for the stochastic enrichment. This means that $sPBC$ has neither refinement nor recursion nor iteration operations. Its denotational semantics was defined in terms of a subclass of labeled continuous time stochastic PNs (LCTSPNs), based on CTSPNs [6, 91] and called stochastic Petri boxes (s-boxes). Calculus $sPBC$ has an interleaving operational semantics in terms of transition systems labeled with multiactions and their rates. In [90], a computing system with n parallel processes and a critical section, as well as the producer/consumer system with a producer, a consumer and a buffer of capacity 1 or n, moreover, the alternating bit protocol with an emitter, a receptor and 2 channels, were described within $sPBC$. Current research in this branch has an aim to extend the specification abilities of $sPBC$ and to define appropriate congruence relation over algebraic formulas. The results on constructing the iteration for $sPBC$ were reported and the producer/consumer system with a buffer of capacity 1 or n was specified in [105,106]. In $sPBC$ with iteration, performance of the processes is evaluated by analyzing their underlying continuous time Markov chains (CTMCs). In the papers [104,107], a number of new equivalence relations were proposed for regular terms of $sPBC$ with iteration to choose later a suitable candidate for a congruence. In [111], special immediate multiactions with zero time delay were added to $sPBC$, and a manufacturing system with 3 machines and an assembler, as well as the AUY-protocol with a sender, receiver and 2 channels, were modeled. 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 SPNs (LGSPNs), based on GSPNs [6, 7, 91] and called generalized stochastic Petri boxes (gs-boxes). The performance analysis in $gsPBC$ is based on the underlying semi-Markov chains (SMCs). Note that the example systems considered within $sPBC$ and its extensions had an interleaving semantics. The performance indices were calculated only for the systems from [90, 111].

To specify mobile systems, a concept of ambient is introduced. An ambient extension of PBC, called Ambient Petri box calculus ($APBC$), was proposed in [54]. Ambient calculus is used to model behaviour of mobile systems. Ambient is a named environment delimited by a boundary. The ambients can be moved to a new location thus modeling mobility. The algebra $APBC$ includes ambients and mobility capabilities. Hence, it could be interpreted as an extension of the Ambient Calculus with the operations of PBC. Basic actions of $APBC$ are capabilities defined over ambient names and standard multiactions of PBC. Only finite part of PBC was taken for the ambient enrichment. Moreover, only concurrency and sequence were transferred into $APBC$ from the set of PBC operations in [54]. This reduced algebra was called Simple Ambient Petri box calculus ($SAPBC$). A denotational semantics was defined in terms of Elementary Object Systems (EOSs) that are two-level net systems composed from a system net and object nets. Object nets could be interpreted as high-level tokens of the system net modeling the execution of mobilie processes. The calculus $SAPBC$ has a step operational semantics in terms of labeled transition systems.

PBC has a step operational semantics, whereas $sPBC$ has an interleaving one. Remember that in step semantics, parallel executions of activities (steps) are permitted while in interleaving semantics, we can execute only single activities. Hence, a stochastic extension of PBC with a step semantics is needed to keep the concurrency degree of behavioural analysis at the same level as in PBC. As mentioned in [97, 99], in contrast to continuous time approach (used in $sPBC$), discrete time approach allows for constructing models of common clock systems and clocked devices. In such models, multiple transition firings (or executions of multiple activities) at time moments (ticks of the central clock) are possible, resulting in a step semantics. Moreover, employment of discrete stochastic time fills the gap between the models with deterministic (fixed) time delays and those with continuous stochastic time delays. As argued in [1], arbitrary delay distributions are much easier to handle in a discrete time domain. In [89, 115, 116], discrete stochastic time was preferred to enable simultaneous expiration of multiple delays. Nevertheless, there were no stochastic extension of PBC with step semantics until recent times. It can be done with the use of labeled DTSPNs as a semantic area, since discrete time models allow for concurrent action occurrences. The enrichment based on DTSPNs is natural because PBC has a step operational semantics.

A notion of equivalence is very important in formal theory of computing processes and systems. Behavioural equivalences are applied during verification stage both to compare behaviour of systems and reduce their structure. At present time, there exists a great diversity of different equivalence notions for concurrent systems, and their interrelations are well explored in the literature. The most popular and widely used one is bisimulation. Unfortunately, the mentioned behavioural equivalences take into account only functional (qualitative) but not performance (quantitative) aspects of system behaviour. Additionally, the equivalences are often interleaving ones, and they do not respect concurrency. SPAs inherited from timeless PAs a possibility to apply equivalences for comparison of specified processes. Like equivalences for other stochastic models, the relations for SPAs have special requirements due to summation of probabilities. The states from which similar future behaviours start have to be grouped into equivalence classes. The classes form elements of the aggregated state space, and they are defined a posteriori while searching for equivalences on state space of a model. In the case of bisimulation equivalence, from every two bisimilar states, the same actions can be executed, and the subsequent states resulting from execution of an action belong to the same equivalence class. In addition, for both states, the cumulative probabilities to move to the same equivalence class by executing the same action coincide. A different kind of quantitative relations are called Markovian equivalences, which take rate (the parameter of exponential distribution that governs time delays) instead of probability. Note that the probabilistic equivalences can be seen as discrete time analogues of the Markovian ones, since the latter are defined as the continuous time relations. The non-interleaving bisimulation equivalence in GSMPA [9, 34] uses ST-semantics for action particles while in Sπ [124] it is based on a sophisticated labeling.

Interleaving probabilistic weak trace equivalence was proposed in [43] on labeled probabilistic transition systems and in [151] it was defined on labeled CTMCs. Interleaving probabilistic strong bisimulation equivalence was proposed in [33, 87, 88] on labeled probabilistic transition systems. Interleaving Markovian strong bisimulation equivalence was constructed in [70] for $MTIPP$, in [64] for $PEPA$ and in [23] for $EMPA$. The mentioned equivalence relation for $PEPA$ has been proved to be a congruence. Interleaving probabilistic equivalences were defined for probabilistic processes in [59, 71]. Interleaving Markovian weak bisimulation equivalence was introduced in [39] on Markovian process algebras, in [40] on stochastic automata, in [41] on labeled CTSPNs and in [42] on GSPNs. Interleaving probabilistic weak and strong bisimulation equivalences were proposed in [24] on labeled probabilistic transition systems and in [25] they were defined on labeled discrete time Markov chains (DTMCs) and CTMCs. In [27], a notion of interleaving stochastic weak bisimulation equivalence for process terms was introduced. The authors proved that the equivalence is preserved by formula composition within SPAs considered in the paper, i.e. the relation is a congruence. In [21], a comprehensive investigation of a variety of interleaving Markovian trace, test, strong and weak bisimulation equivalences was carried out on sequential and concurrent Markovian process calculi. At the same time, no appropriate equivalence notion was defined for concurrent SPAs so far. Thus, it is desirable to propose an equivalence relation for parallel SPAs that binds formulas specifying processes with similar behavior and differentiates those having non-similar one from a certain viewpoint. It would be fine to find a relation that is a congruence with respect to the algebraic operations. In this case, the formulas combined by algebraic operations from equivalent subformulas will be equivalent as well. This is very important property while bottom-up design of processes.

We did some work on the development of concurrent discrete time SPNs and SPAs, as well as on defining a variety of concurrent probabilistic equivalences. In [36, 37], labeled weighted DTSPNs (LWDTSPNs) were proposed that is a modification of DTSPNs [99] by transition labeling and weights. In [38, 134], labeled DTSPNs (LDTSPNs) were introduced. Transitions of LWDTSPNs and LDTSPNs are labeled by actions which represent elementary activities and can be visible or invisible to an external observer. For these two net classes, a number of new probabilistic τ-trace and τ-bisimulation equivalences were defined that abstract from invisible actions (denoted by τ) and respect concurrency in different degrees (interleaving and step relations). In addition, probabilistic relations that require back or back-forth simulation were introduced. An application of the probabilistic back-forth τ-bisimulation equivalences to compare the stationary behaviour of the LWDTSPNs or LDTSPNs was demonstrated. In [38, 129], a logical characterization was presented for interleaving and step probabilistic τ-bisimulation equivalences via formulas of the new probabilistic modal logics. The characterization means that two LWDTSPNs or LDTSPNs are (interleaving or step) probabilistic τ-bisimulation equivalent if they satisfy the same formulas of the corresponding probabilistic modal logic. Thus, instead of comparing nets operationally, one have to check the corresponding satisfaction relation only applying standard verification techniques. The new interleaving and step logics are modifications of that, called PML, was proposed in [87] on probabilistic transition systems with visible actions. In [37, 38, 134], a stochastic algebra of finite nondeterministic processes $StAFP_0$ was proposed with semantics in terms of a subclass of LWDTSPNs and LDTSPNs, called stochastic acyclic nets (SANs). The calculus defined is a stochastic extension of the algebra AFP_0 introduced in [78]. $StAFP_0$ specifies concurrent stochastic processes. Another feature of the algebra is a net semantics allowing one to preserve the level of parallelism, since Petri nets is a classical “true concurrency” model. Usually, transition systems are used for this purpose, but they are not able to respect concurrency completely. An axiomatization for the semantic equivalence of $StAFP_0$ was proposed. It was proved that any algebraic formula could be reduced to the “fully stratified” one with the use of the axiom system. This simplifies semantic comparison of formulas. In [130, 134], we considered different classes of stochastic Petri nets. We explored how transition labeling could be defined to compare SPNs by equivalences. An suitability of the SPN classes for modeling and analysis of different kinds of dynamic systems was investigated. In [131, 133], a discrete time stochastic extension $dtsPBC$ of finite PBC was constructed. In $dtsPBC$, the residence time in the process states is geometrically distributed. A step operational and a net denotational semantics of $dtsPBC$ were defined, and their consistency was demonstrated. In addition, 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 [132, 135–144], we constructed an enrichment of $dtsPBC$ with the iteration operator used to specify infinite processes. The performance evaluation in $dtsPBC$ with iteration is accomplished via the underlying discrete time Markov chains (DTMCs) of the algebraic processes. Since $dtsPBC$ has a discrete time semantics and geometrically distributed sojourn time in the process states, unlike $sPBC$ with continuous time semantics and exponentially distributed delays, the calculi apply two different approaches to the stochastic extension of PBC, in spite of some similarity of their syntax and semantics inherited from PBC. The main advantage of $dtsPBC$ is that concurrency is treated naturally, like in PBC, whereas in $sPBC$ parallelism is simulated by interleaving, obliging one to collect the information on causal independence of activities before constructing the semantics. In [145–149], we presented the extension

$dtsiPBC$ of the latter calculus with immediate multiactions. The performance analysis in $dtsiPBC$ is based on the underlying semi-Markov chains (SMCs) and (reduced) DTMCs.

Let us compare $dtsPBC$ with classical SPAs: $MTIPP$ [70], $PEPA$ [64] and $EMPA$ [23]. The first main difference comes from PBC, since $dtsPBC$ is based on this calculus. In particular, all algebraic operations are inherited from PBC, as well as a notion of multiaction. The second main difference is discrete probabilities of activities induced by discrete time semantics, whereas the action rates are used in the standard SPAs with continuous time semantics. Let us explain this all in more detail. In $dtsPBC$, every activity is a pair consisting of the multiaction (not just an action, as in the classical SPAs) and its probability (not the rate, as in the mentioned SPAs) to be executed independently The algebra $dtsPBC$ has the sequence operator in contrast to prefix one in the three SPAs which we compare with. One can combine arbitrary expressions with sequence operation, i.e. it is more flexible than the prefix one, where the first argument should be a single activity. The choice operator in $dtsPBC$ is analogous to that in $MTIPP$ and $PEPA$, as well as to the alternative composition in $EMPA$, in the sense that the choice is probabilistic, but a discrete probability function is used in $dtsPBC$, unlike continuous ones in the mentioned calculi. On the other hand, concurrency and synchronization in $dtsPBC$ are different operations (this feature is inherited from PBC), unlike the situation in the classical SPAs where parallel composition (combinator) has a synchronization capability. Relabeling in $dtsPBC$ is analogous to that in $EMPA$, but it is additionally extended to conjugated actions. The restriction operation in $dtsPBC$ differs from hiding in $PEPA$ and functional abstraction in $EMPA$, where the hidden actions are labeled with a symbol of "silent" action τ, like in $TCSP$. In $dtsPBC$, restriction by an action means that, for a given expression, any process behaviour containing the action or its conjugate is not allowed. The synchronization on an elementary action in $dtsPBC$ collects all the pairs consisting of this elementary action and its conjugate which are contained in the multiactions from the synchronized activities. The operation produces new activities such that the first element of every resulting activity is the union of the multiactions from which all the mentioned pairs of conjugated actions are removed, and the second element is the product of the probabilities of the activities involved in the synchronization. Thus, there is a difference with the way synchronization is applied in the mentioned SPAs where it is accomplished over identical action names, and every resulting activity consists of the same action name and the rate calculated via sums, minimums and products of the rates of the initial activities, such as the apparent rate in $PEPA$. The algebra $dtsPBC$ has no recursion operation or a possibility for recursive definitions, but it includes the iteration operation that gives an ability to specify infinite behaviour with the explicitly defined start and termination. Iteration allows for a syntactic description of many realistic processes with loops. Calculus $dtsPBC$ has a discrete time semantics, and residence time in the states is geometrically distributed, unlike the mentioned SPAs with continuous time semantics and exponentially distributed activity delays. As a consequence, the semantics of $dtsPBC$ is the step one in contrast to the interleaving semantics of the three SPAs mentioned above. The performance can be investigated based on the discrete time Markov chain (DTMC) extracted from the labeled probabilistic transition system associated with each expression of $dtsPBC$. Note that in the classical SPAs we generate CTMCs from the transition systems. In [57], a denotational semantics of $PEPA$ has been proposed via $PEPA$ nets that are high-level CTSPNs with coloured tokens (coloured CTSPNs), from which the underlying CTMCs can be retrieved. In [11, 20], a denotational semantics of EMPA based on GSPNs has been defined, from which one can also extract the underlying SMCs and CTMCs (when both immediate and timed transitions are present) or discrete time Markov chains (DTMCs) (but when there are only immediate transitions). In addition, $dtsPBC$ has a denotational semantics in terms of LDTSPNs from which the corresponding DTMCs can be derived as well.

Only a few non-interleaving SPAs were considered among non-Markovian ones [4, 74]. The semantics of all Markovian calculi is interleaving and their action delays have exponential distribution, which is the only continuous probability distribution with memoryless (Markovian) property. In [28], Generalized Stochastic Process Algebra ($GSPA$) was introduced. It has a true-concurrent denotational semantics in terms of generalized stochastic event structures (GSESs) with non-Markovian stochastic delays of events. In that paper, no operational semantics or performance evaluation methods for $GSPA$ were presented. Later, in [77], generalized semi-Markov processes (GSMPs) were extracted from GSESs to analyze performance. In [123, 124], generalized Stochastic π-calculus ($S\pi$) with general continuous distributions of activity delays was defined. It has a proved operational semantics with transitions labeled by encodings of their deduction trees. No well-established underlying performance model for this version of $S\pi$ was described. In [9, 34], Generalized Semi-Markovian Process Algebra ($GSMPA$) was developed with an ST-operational semantics and non-Markovian action delays. The performance analysis in $GSMPA$ is accomplished via GSMPs. Again, the first fundamental difference between $dtsPBC$ and the calculi $GSPA$, $S\pi$ and $GSMPA$ is that $dtsPBC$ is based on PBC, whereas $GSPA$ is an extension of simple Process Algebra (PA) from [28], $S\pi$ extends π-calculus [100, 101] and $GSMPA$ is an enrichment of $EMPA$. Therefore, both $GSPA$ and $GSMPA$ have prefixing, choice (alternative composition), parallel composition, renaming (relabeling) and hiding (abstraction) operations, but only $GSMPA$ permits constants. Unlike $dtsPBC$, $GSPA$ has neither iteration or recursion, $GSMPA$ allows only recursive definitions, whereas $S\pi$ additionally has operations to specify mobility. Note also that $GSPA$, $S\pi$ and $GSMPA$ do not specify instantaneous events or activities while dtsiPBC has immediate multiactions. The second significant difference is that geometrically distributed delays are associated with process states in $dtsPBC$, unlike generally distributed delays assigned to events in $GSPA$ or to activities in $S\pi$ and $GSMPA$. As a consequence, $dtsPBC$ has a discrete time operational semantics allowing for concurrent execution of activities in steps. $GSPA$ has no operational semantics while $S\pi$ and $GSMPA$ have continuous time ones. In continuous time semantics, concurrency is simulated by interleaving, since simultaneous occurrence of any two events has zero probability according to the properties of continuous probability distributions. Therefore, interleaving transitions are often annotated with an additional information to keep concurrency data. The transition labels in the operational semantics of $S\pi$ encode the action causality information and allow one to derive the enabling relations and the firing distributions of concurrent transitions from the transition sequences. At the same time, abstracting from stochastic delays leads to the classical early interleaving semantics of π-calculus. The ST-operational semantics of $GSMPA$ is based on decorated transition systems governed by transition rules with rather complex preconditions. There are two types of transitions: the choice (action beginning) and the termination (action ending) ones. The choice transitions are labeled by weights of single actions chosen for execution while the termination transitions have no labels. Only single actions can begin, but several actions can end in parallel. Thus, the choice transitions happen just sequentially while the termination transitions can happen simultaneously. As a result, the decorated interleaving / step transition systems are obtained. $dtsPBC$ has an SPN-based denotational semantics. In comparison with event structures, PNs are more expressive and visually tractable formalism, capable of finitely specifying an infinite behaviour. Recursion in $GSPA$ produces infinite GSESs while $dtsPBC$ has iteration operation with a finite SPN semantics. An identification of infinite GSESs that can be finitely represented in $GSPA$ was left for a future research.

In [1], a class of compositional DTSPNs with generally distributed discrete time transition delays was proposed, called dts-nets. The denotational semantics of a stochastic extension (we call it stochastic ACP or $sACP$) of a subset of Algebra of Communicating Processes (ACP) [26] can be constructed via dts-nets. There are two types of transitions in dts-nets: immediate (timeless) ones, with zero delays, and time ones, whose delays are random variables having general discrete distributions. The top-down synthesis of dts-nets consists in the substitution of their transitions by blocks (dts-subnets) corresponding to the sequence, choice, parallelism and iteration operators. It was explained how to calculate the throughput time of dts-nets using the service time (defined as holding time or delay) of their transitions. For this, the notions of service distribution for the transitions and throughput distribution for the building blocks were defined. Since the throughput time of the parallelism block was calculated as the maximal service time for its two constituting transitions, the analogue of the step semantics approach was implemented. In [115, 116], an SPA called Theory of Communicating Processes with discrete stochastic time (TCP^{dst}) was introduced, later in [89] called Theory of Communicating Processes with discrete real and stochastic time (TCP^{drst}). It has discrete real time (deterministic) delays (including zero time delays) and discrete stochastic time delays. The algebra generalizes real time processes to discrete stochastic time ones by applying real time properties to stochastic time and imposing race condition to real time semantics. TCP^{dst} has an interleaving operational semantics in terms of stochastic transition systems. The performance is analyzed via discrete time probabilistic reward graphs which are essentially the reward transition systems with probabilistic states having finite number of outgoing probabilistic transitions and timed states having a single outgoing timed transition. The mentioned graphs can be transformed by unfolding or geometrization into discrete time Markov reward chains (DTMRCs) appropriate for transient or long-run (stationary) analysis. The first difference between $dtsPBC$ and the algebras $sACP$ and TCP^{dst} is that $dtsPBC$ is based on PBC, but $sACP$ and TCP^{dst} are the extensions of ACP. $sACP$ has taken from ACP only sequence, choice, parallelism and iteration operations, whereas $dtsPBC$ has additionally relabeling, restriction and synchronization ones, inherited from PBC. In TCP^{dst}, besides standard action prefixing, alternative, parallel composition, encapsulation (similar to restriction) and recursive variables, there are also timed delay prefixing, dependent delays scope and the maximal time progress operators, which are new both for ACP and $dtsPBC$. The second difference is that $dtsiPBC$, $sACP$ and TCP^{dst}, all have zero delays, however, discrete time delays in $dtsiPBC$ are zeros or geometrically distributed and associated with process states. The zero delays are possible just in vanishing states while geometrically distributed delays are possible only in tangible states. For each tangible state, the parameter of geometric distribution governing the delay in the state is completely determined by the probabilities of all stochastic multiactions executable from it. In $sACP$ and TCP^{dst}, delays are generally distributed, but they are assigned to transitions in $sACP$ and separated from actions (excepting zero delays) in TCP^{dst}. Moreover, a special attention is given to zero delays in $sACP$ and deterministic delays in TCP^{dst}. In $sACP$, immediate (timeless) transitions with zero delays serve as source and sink transitions of the dts-subnets corresponding to the choice, parallelism and iteration operators. In TCP^{dst}, zero delays of actions are specified by undelayable action prefixes while positive deterministic delays of

Table 1: Classification of stochastic process algebras

Time	Interleaving semantics	Non-interleaving semantics
Continuous	$MTIPP$ (CTMC), $\boldsymbol{PEPA}$ (CTMP), $\boldsymbol{EMPA}$ (SMC, CTMC), $\boldsymbol{sPBC}$ (CTMC), $\boldsymbol{gsPBC}$ (SMC)	$GSPA$ (GSMP), $S\pi$, $GSMPA$ (GSMP)
Discrete	TCP^{dst} (DTMRC)	$\boldsymbol{sACP}$, $\boldsymbol{dtsPBC}$ (DTMC), $\boldsymbol{dtsiPBC}$ (SMC, DTMC)

processes are specified with timed delay prefixes. Neither formal syntax nor operational semantics for $sACP$ are defined and it is not explained how to derive Markov chains from the algebraic expressions or the corresponding dts-nets to analyze performance. It is not stated explicitly, which type of semantics (interleaving or step) is accommodated in $sACP$. In spite of the discrete time approach, operational semantics of TCP^{dst} is still interleaving, unlike that of $dtsPBC$. In addition, no denotational semantics was defined for TCP^{dst}.

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

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

Moreover, the existence of an analytical solution allows one to interpret quantitative values (rates, probabilities, weights etc.) from the system specifications as parameters, which can be adjusted to optimize the system performance, like in $dtsPBC$ [142,143], $dtsiPBC$ [146,148] and parametric probabilistic transition systems (i.e. DTMCs whose transition probabilities may be real-value parameters) [86]. Note that DTMCs whose transition probabilities are parameters were introduced in [45]. Parametric CTMCs with the transition rates treated as parameters were investigated in [65]. On the other hand, no parameters in formulas of SPAs were considered in the literature so far. In dtsPBC we can easily construct examples with more parameters than we did in our case study. The performance indices will be then interpreted as functions of several variables. The advantage of our approach is that, unlike of the method from [86], we should not impose to the parameters any special conditions needed to guarantee that the real values, interpreted as the transition probabilities, always lie in the interval $[0;1]$. To be convinced of this fact, just remember that, as we have demonstrated, the positive probability functions PF, PT, PM, PM^* define probability distributions, hence, they always return values belonging to $(0;1]$ for any probability parameters from $(0;1)$. In addition, the transition constraints (their probabilities, rates and guards), calculated using the parameters, in our case should not always be polynomials over variables-parameters, as often required in the mentioned papers, but they may also be fractions of polynomials, like in the case studies that we have constructed.

One can see that the stochastic process calculi proposed in the literature are based on interleaving, as a rule, and parallelism is simulated by synchronous or asynchronous execution. As a semantic domain, the interleaving formalism of transition systems is often used. However, to properly support intuition of the behaviour of concurrent and distributed systems, their semantics should treat parallelism as a primitive concept that cannot be reduced to nondeterminism. Moreover, in interleaving semantics, some important properties of such systems cannot be expressed, such as simultaneous occurrence of concurrent transitions [50] or local deadlock in the spatially distributed processes [102]. Therefore, investigation of stochastic extensions for more expressive and powerful algebraic calculi is an important issue. The development of step or “true concurrency” (such that parallelism is considered as a causal independence) SPAs is an interesting and nontrivial problem, which has attracted special attention last years. Nevertheless, not so many formal stochastic models of parallel systems were defined whose underlying stochastic processes are based on DTMCs. As mentioned in [56], such models are more difficult to analyze, since several events can occur simultaneously in discrete time systems (the models have a step semantics) and the probability of a set of events cannot be easily related to the probability of the single ones. Therefore, parallel executions of actions are often not considered also in the discrete time SPAs, such as TCP^{dst}, whose underlying stochastic process is DTMCs with rewards (DTMRCs). As observed in [68, 69], even for stochastic models with generally distributed time delays, some restrictions on the concurrency degree were imposed to simplify their analysis techniques. In particular, the enabling restriction requires that no two generally distributed transitions are enabled in any reachable marking. Hence, their activity periods do not intersect and no two such transitions can fire simultaneously, this results in interleaving semantics of the model.

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

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

Thus, the main advantages of $dtsPBC$ are the flexible multiaction labels and the set of very powerful operations, as well as a step operational and a Petri net denotational semantics allowing for really concurrent execution of activities (or transitions), together with an ability for analytical and parametric performance evaluation. The uniqueness of our approach consists in applying a parallel semantics for the process expressions and preserving the concurrency level in the extracted performance model (DTMC) through its state changes corresponding to the simultaneous executions. The salient point of $dtsPBC$ is a combination of discrete stochastic time and step semantics in an SPA.

In this paper, we investigate equivalence notions for $dtsPBC$ with iteration. First, we present the syntax of the extended $dtsPBC$. Each multiaction of the initial calculus PBC is associated with a probability. Such a pair is called stochastic multiaction or activity. Second, we propose semantics of $dtsPBC$. The step operational semantics is constructed in terms of labeled probabilistic transition systems based on action and inaction rules. The denotational semantics is defined in terms of a subclass of LDTSPNs, based on DTSPNs [99] and called discrete time stochastic Petri boxes (dts-boxes). Consistency of operational and denotational semantics is proved. Further, we define a number of stochastic equivalences in the algebraic setting based on transition systems without empty behaviour. These relations are weaker than the semantic equivalence of $dtsPBC$. They are used to identify stochastic processes with similar behaviour which are differentiated by the semantic equivalence that is too discriminate in many cases. The interrelations diagram of all the introduced equivalences is built. We describe how the stochastic equivalences can be used to reduce transition systems of expressions and the related formalisms. We present a characterization of the stochastic bisimulation equivalences via two new probabilistic modal logics based on PML. It is demonstrated how to compare stochastic processes in their steady states with the use of the relations. Moreover, a problem of preservation of the equivalence notions by algebraic operations is discussed. The proposed equivalences are used to construct a congruence relation. At the end, we present two case studies explaining how to analyze performance of systems within the calculus. We consider algebraic models of shared memory system and dining philosophers one. Thus, the main contributions of the paper are the following.

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

The paper is organized as follows. In the next Section 2, the syntax of the extended calculus $dtsPBC$ is presented. Then, in Section 3, we construct the operational semantics of the algebra in terms of labeled transition systems. In Section 4, we propose the denotational semantics based on a subclass of LDTSPNs. Section 5 is devoted to the construction and the interrelations of the stochastic algebraic equivalences based on transition systems without empty loops. In Section 6 we explain how one can reduce transition systems and the related formalisms modulo the equivalences. A logical characterization of the equivalences is presented in Section 7. In Section 8, an application of the relations to comparison of the stationary behaviour is investigated. Preservation of the equivalences by the algebraic operations, i.e. a congruence problem is discussed in Section 9. Section 10 contains two examples of performance evaluation for systems specified by the algebraic expressions. The concluding Section 11 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 of the discrete time stochastic extension of finite PBC enriched with iteration, called *discrete time stochastic Petri box calculus* ($dtsPBC$).

First, we recall a definition of multiset that is an extension of the set notion by allowing several identical elements.

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

We denote the *set of all finite multisets* over a set X by $I\!N_{fin}^X$. Let $M, M' \in I\!N_{fin}^X$. The *cardinality* of M is defined as $|M| = \sum_{x \in X} M(x)$. We write $x \in M$ if $M(x) > 0$ and $M \subseteq M'$ if $\forall x \in X\ M(x) \leq M'(x)$. 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 interpreted as a proper set and denoted by $M \subseteq X$. 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 $\mathcal{L} = I\!N_{fin}^{\mathcal{A}}$ be the set of *all multiactions*. Note that $\emptyset \in \mathcal{L}$, this corresponds to an internal move, i.e. the execution of a multiaction that contains no visible action names. The *alphabet* of $\alpha \in \mathcal{L}$ is defined as $\mathcal{A}(\alpha) = \{x \in \mathcal{A} \mid \alpha(x) > 0\}$.

An *activity (stochastic multiaction)* is a pair (α, ρ), where $\alpha \in \mathcal{L}$ and $\rho \in (0; 1)$ is the *probability* of the multiaction α. This probability is interpreted as that of independent execution of the stochastic multiaction at the next discrete time moment. Such probabilities are used to calculate those to execute (possibly empty) multisets of stochastic multiactions after one time unit delay. The probabilities of stochastic multiactions are required not to be equal to 1 to avoid extra model complexity, since in this case one should assign with them weights, needed to make a choice when several stochastic multiactions with probability 1 can be executed from a state. In this case, some problems appear with conflicts resolving. See [99] for the discussion on SPNs. This decision also allows us to avoid technical difficulties related to conditioning events with probability 0. Another reason is that not allowing probability 1 for stochastic multiactions excludes a potential source of periodicity (hence, non-ergodicity) in the underlying DTMCs of the algebraic expressions. In this version of the algebra, we do not allow instantaneous multiactions. On the other hand, there is no sense to allow zero probabilities of stochastic multiactions, since they would never be performed in this case. Let $\mathcal{SL}$ be the set of *all activities*. Let us note that the same multiaction $\alpha \in \mathcal{L}$ may have different probabilities in the same specification.

The *alphabet* of an activity $(\alpha, \rho) \in \mathcal{SL}$ is defined as $\mathcal{A}(\alpha, \rho) = \mathcal{A}(\alpha)$. The *alphabet* of a multiset of activities $\Gamma \in I\!N_{fin}^{\mathcal{SL}}$ is defined as $\mathcal{A}(\Gamma) = \cup_{(\alpha,\rho)\in\Gamma}\mathcal{A}(\alpha)$. For an activity $(\alpha, \rho) \in \mathcal{SL}$, we define its *multiaction part* as $\mathcal{L}(\alpha, \rho) = \alpha$ and its *probability part* as $\Omega(\alpha, \rho) = \rho$. The *multiaction part* of a multiset of activities $\Gamma \in I\!N_{fin}^{\mathcal{SL}}$ is defined as $\mathcal{L}(\Gamma) = \sum_{(\alpha,\rho)\in\Gamma} \alpha$. Remember that sums and products are considered with the multiplicity when applied to multisets.

Activities are combined into formulas (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 (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* [96].

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 a usual way: for $\alpha \in \mathcal{L}$ we define $f(\alpha) = \sum_{x\in\alpha} f(x)$. Relabeling is extended to activities: for $(\alpha, \rho) \in \mathcal{SL}$, we define $f(\alpha, \rho) = (f(\alpha), \rho)$. Relabeling is extended to the multisets of activities as follows: for $\Gamma \in I\!N_{fin}^{\mathcal{SL}}$ we define $f(\Gamma) = \sum_{(\alpha,\rho)\in\Gamma}(f(\alpha), \rho)$.

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

Let $\alpha, \beta \in \mathcal{L}$ be two multiactions such that for some elementary action $a \in Act$ we have $a \in \alpha$ and $\hat{a} \in \beta$, or $\hat{a} \in \alpha$ and $a \in \beta$. Then synchronization of α and β by a is defined as $\alpha \oplus_a \beta = \gamma$, where

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

In other words, we require that $\alpha \oplus_a \beta = \alpha + \beta - \{a, \hat{a}\}$, i.e. we remove one exemplar of a and one exemplar of $\hat{a}$ from the multiset sum $\alpha + \beta$, since the synchronization of a and $\hat{a}$ produces $\emptyset$. Activities are synchronized with the use of 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 $\alpha \oplus_a \beta$. 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 process 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.

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

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

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

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

To make the grammar above unambiguous, one can add parentheses in the productions with binary operations: $(E; E)$, $(E[]E)$, $(E\|E)$ or to associate priorities with operations. However, here and further we prefer the *PBC* approach: we add parentheses to resolve ambiguities when needed and we assume no priorities.

To avoid technical difficulties with the iteration operator, we should not allow any concurrency at the highest level of the second argument of iteration. This is not a severe restriction though, since we can always prefix parallel expressions by an activity with the empty multiaction part. Later on, in Example 4.4, we shall demonstrate that relaxing the restriction can result in nets which are not safe. Alternatively, we can use a different, safe, version of the iteration operator, but its net translation has six arguments. See also [15] for discussion on this subject. Remember that a PN is *n-bounded* ($n \in I\!N$) if for all its reachable (from the initial marking by the sequences of transition firings) markings there are at most n tokens in every place, and a PN is *safe* if it is 1-bounded.

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

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

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

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

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

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

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

Note that if the underlying static expression of a dynamic one is not regular, the corresponding LDTSPN can be non-safe (though, it is 2-bounded in the worst case [15]).

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

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

3 Operational semantics

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

3.1 Inaction rules

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

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

First, in Table 2, we define inaction rules for the regular dynamic expressions in the form of overlined and underlined regular static ones. In this table, $E, F, K \in RegStatExpr$ and $a \in Act$.

Table 2: Inaction rules for overlined and underlined regular static expressions

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

Second, in Table 3, we introduce inaction rules for the regular dynamic expressions in the arbitrary form. In this table, $E, F \in RegStatExpr$, $G, H, \widetilde{G}, \widetilde{H} \in RegDynExpr$ and $a \in Act$.

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

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

Note that any regular dynamic expression can be always transformed into a (not necessarily unique) operative one by using the inaction rules.

In the following, we shall consider regular expressions only, hence, we can omit the word "regular".

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

Table 3: Inaction rules for arbitrary regular dynamic expressions

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

3.2 Action and empty loop rules

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

The action rules describe dynamic expression transformations in the form of $G \stackrel{\Gamma}{\rightarrow} \widetilde{G}$ due to execution of non-empty multisets Γ of activities. The rules represent possible state changes of the specified processes when some non-empty multisets of activities are executed. As we shall see, the application of an action rule to a dynamic expression leads to a discrete time tick in the corresponding LDTSPN at which some transitions fire and the current marking is possibly changed The current marking remains unchanged only if there is a self-loop produced by the iterative execution of a non-empty multiset, which must be one-element, i.e. the single stochastic (or immediate) multiaction. The reason is the regularity requirement that allows no concurrency at the highest level of the second argument of iteration.

The empty loop rule describes dynamic expression transformations in the form of $G \stackrel{\emptyset}{\rightarrow} G$ due to execution of the empty multiset of activities at a discrete time tick. The rule reflects a non-zero probability to stay in the current state at the next time moment, which is an essential feature of discrete time stochastic processes. As we shall see, the application of the empty loop rule to a dynamic expression leads to a discrete time tick in the corresponding LDTSPN at which no transitions fire and the current marking is not changed. This is a new rule that has no prototype among inaction rules of PBC, since it represents a time delay, but no notion of time exists in PBC. The PBC rule $G \stackrel{\emptyset}{\rightarrow} G$ from [15] in our setting would correspond to the rule $G \Rightarrow G$ that describes staying in the current state when no time elapses. Since we do not need the latter rule to transform dynamic expressions into operative ones and it can even destroy the definition of operative expressions, we do not introduce it in $dtsiPBC$.

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

In Table 4, we define the action and empty loop rules. In the table, $(\alpha, \rho), (\beta, \chi) \in \mathcal{SL},\ E, F \in RegStatExpr$, $G, H \in OpRegDynExpr,\ \widetilde{G}, \widetilde{H} \in RegDynExpr$ and $a \in Act$. Moreover, $\Gamma, \Delta \in I\!N_{fin}^{\mathcal{SL}} \setminus \{\emptyset\}$ and $\Gamma' \in I\!N_{fin}^{\mathcal{SL}}$.

We use the following abbreviations in the names of the rules from the table: "**El**" for "**E**mpty **l**oop", **B**" for "**B**asis case", **SC**" for "**S**equence and **C**hoice", **P**" for "**P**arallel", **L**" for "re**L**abeling", **Rs**" for "**R**e**s**triction", **I**" for "**I**teraton" and **Sy**" for "**Sy**nchronization". The first rule in the table is the empty loop rule **El**. The other rules are the action rules, describing transformations of dynamic expressions, which are built using particular algebraic operations.

Almost all the rules in Table 4 (excepting **El**, **P3** and **Sy2**) resemble those of $sPBC$ [112], but the former correspond to execution of multisets of activities, not of single activities, as in the latter.

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

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

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

Rule **Sy2** establishes that the synchronization of two stochastic multiactions is made by taking the product of their probabilities, since we are considering that both must occur for the synchronization to happen, so

Table 4: Action and empty loop rules

El $G \xrightarrow{\emptyset} G$	**B** $\dfrac{}{\overline{(\alpha,\rho)} \xrightarrow{\{(\alpha,\rho)\}} \underline{(\alpha,\rho)}}$	**SC1** $\dfrac{G \xrightarrow{\Gamma} \widetilde{G},\ \circ \in \{;, []\}}{G \circ E \xrightarrow{\Gamma} \widetilde{G} \circ E}$
SC2 $\dfrac{G \xrightarrow{\Gamma} \widetilde{G},\ \circ \in \{;, []\}}{E \circ G \xrightarrow{\Gamma} E \circ \widetilde{G}}$	**P1** $\dfrac{G \xrightarrow{\Gamma} \widetilde{G}}{G \parallel H \xrightarrow{\Gamma} \widetilde{G} \parallel H}$	**P2** $\dfrac{H \xrightarrow{\Gamma} \widetilde{H}}{G \parallel H \xrightarrow{\Gamma} G \parallel \widetilde{H}}$
P3 $\dfrac{G \xrightarrow{\Gamma} \widetilde{G},\ H \xrightarrow{\Delta} \widetilde{H}}{G \parallel H \xrightarrow{\Gamma+\Delta} \widetilde{G} \parallel \widetilde{H}}$	**L** $\dfrac{G \xrightarrow{\Gamma} \widetilde{G}}{G[f] \xrightarrow{f(\Gamma)} \widetilde{G}[f]}$	**Rs** $\dfrac{G \xrightarrow{\Gamma} \widetilde{G},\ a, \hat{a} \notin \mathcal{A}(\Gamma)}{G \text{ rs } a \xrightarrow{\Gamma} \widetilde{G} \text{ rs } a}$
I1 $\dfrac{G \xrightarrow{\Gamma} \widetilde{G}}{[G * E * F] \xrightarrow{\Gamma} [\widetilde{G} * E * F]}$	**I2** $\dfrac{G \xrightarrow{\Gamma} \widetilde{G}}{[E * G * F] \xrightarrow{\Gamma} [E * \widetilde{G} * F]}$	**I3** $\dfrac{G \xrightarrow{\Gamma} \widetilde{G}}{[E * F * G] \xrightarrow{\Gamma} [E * F * \widetilde{G}]}$
Sy1 $\dfrac{G \xrightarrow{\Gamma} \widetilde{G}}{G \text{ sy } a \xrightarrow{\Gamma} \widetilde{G} \text{ sy } a}$	**Sy2** $\dfrac{G \text{ sy } a \xrightarrow{\Gamma' + \{(\alpha,\rho)\} + \{(\beta,\chi)\}} \widetilde{G} \text{ sy } a,\ a \in \alpha,\ \hat{a} \in \beta}{G \text{ sy } a \xrightarrow{\Gamma' + \{(\alpha \oplus_a \beta, \rho \cdot \chi)\}} \widetilde{G} \text{ sy } a}$	

this corresponds, in some sense, to the probability of the independent event intersection, but the real situation is more complex, since these stochastic multiactions can also be executed in parallel. Nevertheless, when scoping (the combined operation consisting of synchronization followed by restriction over the same action [15]) is applied over a parallel execution, we get as final result just the simple product of the probabilities, since no normalization is needed there. Multiplication is an associative and commutative binary operation that is distributive over addition, i.e. it fulfills all practical conditions imposed on the synchronization operator in [63]. Further, if both arguments of multiplication are from $(0;1)$ then the result belongs to the same interval, hence, multiplication naturally maintains probabilistic compositionality in our model. Our approach is similar to the multiplication of rates of the synchronized actions in $MTIPP$ [70] in the case when the rates are less than 1. Moreover, for the probabilities ρ and χ of two stochastic multiactions to be synchronized we have $\rho \cdot \chi < \min\{\rho, \chi\}$, i.e. multiplication meets the performance requirement stating that the probability of the resulting synchronized stochastic multiaction should be less than the probabilities of the two ones to be synchronized. While performance evaluation, it is usually supposed that the execution of two components together require more system resources and time than the execution of each single one. This resembles the *bounded capacity* assumption from [63]. Thus, multiplication is easy to handle with and it satisfies the algebraic, probabilistic, time and performance requirements. Therefore, we have chosen the product of the probabilities for the synchronization. See also [28, 35] for a discussion about binary operations producing the rates of synchronization in the continuous time setting.

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

Observe also that we do not allow a self-synchronization, i.e. synchronization of an activity with itself. The purpose of this restriction is to avoid rather cumbersome and unexpected behaviour, as well as many technical difficulties [15].

In Table 5, inaction rules, action rules and empty loop rule are compared according to the three questions about their application: whether it changes the current state, whether it leads to a time progress, and whether it results in execution of some activities. Positive answers to the questions are denoted by the plus sign while negative ones are specified by the minus sign. If both positive and negative answers can be given to some of the questions in different cases then the plus-minus sign is written. Notice that the process states are considered up to structural equivalence of the corresponding expressions, and time progress is not regarded as a state change.

3.3 Transition systems

We now intend to construct labeled probabilistic transition systems associated with dynamic expressions. The transition systems will be used to define the operational semantics of expressions of $dtsPBC$.

Table 5: Comparison of inaction, action and empty loop rules

Rules	State change	Time progress	Activities execution
Inaction rules	$-$	$-$	$-$
Action rules	$\pm$	$+$	$+$
Empty loop rule	$-$	$+$	$-$

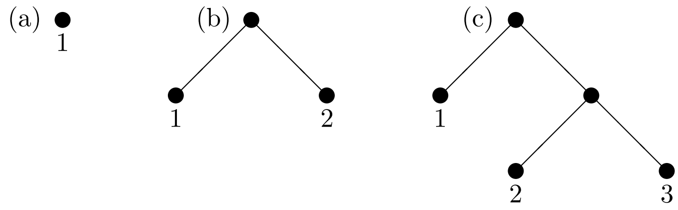

Figure 1: The binary trees encoded with the numberings 1, (1)(2) and (1)((2)(3))

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

Definition 3.3 *The* numbering *of expressions is defined as*

$$\iota ::= \; n \mid (\iota)(\iota),$$

where $n \in I\!N$.

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

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

The new activities resulting from applications of the second rule for synchronization **Sy2** in different orders should be considered up to permutation of their numbering. In this way, we shall recognize different instances of the same activity. If we compare the contents of different numberings, i.e. the sets of natural numbers in them, we shall be able to identify the mentioned instances.

The *content* of a numbering $\iota \in Num$ is

$$Cont(\iota) = \begin{cases} \{\iota\}, & \iota \in I\!N; \\ Cont(\iota_1) \cup Cont(\iota_2), & \iota = (\iota_1)(\iota_2). \end{cases}$$

After we apply the enumeration, the multisets of activities from the expressions become the proper sets. In the following, we suppose that the identical activities are enumerated when needed to avoid ambiguity. This enumeration is considered to be implicit.

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* $X/_{\mathcal{E}} = \{[x]_{\mathcal{E}} \mid x \in X\}$.

Definition 3.4 *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. 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* $\exists \Gamma \; H \xrightarrow{\Gamma} \widetilde{H}$ *then* $[\widetilde{H}]_{\approx} \in DR(G)$.

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

The set of *all multisets of activities executable in* s is defined as $Exec(s) = \{\Gamma \mid \exists H \in s\ \exists \widetilde{H}\ H \stackrel{\Gamma}{\rightarrow} \widetilde{H}\}$.

Note that if $\Gamma \in Exec(s)$ then by rules **P3**, **Sy2** and definition of $Exec(s)$ we have $\forall \Delta \subseteq \Gamma\ \Delta \in Exec(s)$, i.e. $2^{\Gamma} \subseteq Exec(s)$.

Since the inaction rules only distribute and move upper and lower bars along the syntax of dynamic expressions, all $H \in s$ have the same underlying static expression F. Process expressions always have a finite length, hence, the number of all (enumerated) activities and the number of all operations in the syntax of F are finite as well. The action rule **Sy2** is the only one that generates new activities. They result from the handshake synchronization of actions and their conjugates belonging to the multiaction parts of the first and second constituent activity, respectively. Since we have a finite number of operators sy in F and all the multiaction parts of the activities are finite multisets, the number of the new synchronized activities is also finite. The action rules contribute to $Exec(s)$ (in addition to the empty set, if rule **El** is applicable) only the sets consisting both of activities from F and the new activities, produced by **Sy2**. Since we have a finite number n of all such activities, we get $|Exec(s)| \leq 2^n < \infty$. Thus, summation and multiplication by elements from the finite set $Exec(s)$ are well-defined.

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

$$PF(\Gamma, s) = \prod_{(\alpha,\rho)\in\Gamma} \rho \cdot \prod_{\{\{(\beta,\chi)\}\in Exec(s) \mid (\beta,\chi)\notin\Gamma\}} (1-\chi).$$

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

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

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

Note that the definition of $PF(\Gamma, s)$ (as well as the definitions of other probability functions which we shall present) is based on the enumeration of activities which is considered implicit.

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

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

Thus, $PT(\Gamma, s)$ can be interpreted as the *conditional* probability to execute Γ in s calculated using the conditional probability formula in the form $\mathsf{P}(Z|W) = \frac{\mathsf{P}(Z \cap W)}{\mathsf{P}(W)}$. The event Z consists in the exclusive execution of Γ in s, hence, $\mathsf{P}(Z) = PF(\Gamma, s)$. The event W consists in the exclusive execution of any multiset (including the empty one) $\Delta \in Exec(s)$ in s. Thus, $W = \cup_j Z_j$, where $\forall j,\ Z_j$ are mutually exclusive events (in a probability sense, i.e. intersection of these events is the empty event) and $\exists i,\ Z = Z_i$. We have $\mathsf{P}(W) =$

Table 6: Calculation of the probability functions PF, PT, PM for $s_1 \in DR(\overline{E})$ and $E = (\{a\},\rho)[](\{a\},\chi)$

$s_1\backslash\Gamma$	$\emptyset$	$\{(\{a\},\rho)\}$	$\{(\{a\},\chi)\}$	Σ
PF	$(1-\rho)(1-\chi)$	$\rho(1-\chi)$	$\chi(1-\rho)$	$1-\rho\chi$
PT	$\frac{(1-\rho)(1-\chi)}{1-\rho\chi}$	$\frac{\rho(1-\chi)}{1-\rho\chi}$	$\frac{\chi(1-\rho)}{1-\rho\chi}$	1
PM	$\frac{(1-\rho)(1-\chi)}{1-\rho\chi}$ (s_1)	$\frac{\rho+\chi-2\rho\chi}{1-\rho\chi}$ (s_2)		1

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

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

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

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

The summation in the definition above reflects the probability of the mutually exclusive event union, since $\sum_{\{\Gamma\mid\exists H\in s,\ \exists\widetilde{H}\in\tilde{s},\ H\xrightarrow{\Gamma}\widetilde{H}\}} PT(\Gamma,s) = \frac{1}{\sum_{\Delta\in Exec(s)} PF(\Delta,s)}\cdot\sum_{\{\Gamma\mid\exists H\in s,\ \exists\widetilde{H}\in\tilde{s},\ H\xrightarrow{\Gamma}\widetilde{H}\}} PF(\Gamma,s)$, where for each Γ, $PF(\Gamma,s)$ is the probability of the exclusive execution of Γ in s. Note that $\forall s\in DR(G)$ $\sum_{\{\tilde{s}\mid\exists H\in s\ \exists\widetilde{H}\in\tilde{s}\ \exists\Gamma\ H\xrightarrow{\Gamma}\widetilde{H}\}} PM(s,\tilde{s}) = \sum_{\{\tilde{s}\mid\exists H\in s\ \exists\widetilde{H}\in\tilde{s}\ \exists\Gamma\ H\xrightarrow{\Gamma}\widetilde{H}\}}\sum_{\{\Gamma\mid\exists H\in s\ \exists\widetilde{H}\in\tilde{s}\ H\xrightarrow{\Gamma}\widetilde{H}\}} PT(\Gamma,s) = \sum_{\Gamma\in Exec(s)} PT(\Gamma,s) = 1$.

Example 3.2 *Let $E = (\{a\},\rho)[](\{a\},\chi)$. $DR(\overline{E})$ consists of the equivalence classes $s_1 = [\overline{E}]_\approx$ and $s_2 = [\underline{E}]_\approx$. The execution probabilities are calculated as follows. Since $Exec(s_1) = \{\emptyset, \{(\{a\},\rho)\}, \{(\{a\},\chi)\}\}$, we get $PF(\{(\{a\},\rho)\},s_1) = \rho(1-\chi)$, $PF(\{(\{a\},\chi)\},s_1) = \chi(1-\rho)$ and $PF(\emptyset,s_1) = (1-\rho)(1-\chi)$. Then $\sum_{\Delta\in Exec(s_1)} PF(\Delta,s_1) = \rho(1-\chi)+\chi(1-\rho)+(1-\rho)(1-\chi) = 1-\rho\chi$. Thus, $PT(\{(\{a\},\rho)\},s_1) = \frac{\rho(1-\chi)}{1-\rho\chi}$, $PT(\{(\{a\},\chi)\},s_1) = \frac{\chi(1-\rho)}{1-\rho\chi}$ and $PT(\emptyset,s_1) = PM(s_1,s_1) = \frac{(1-\rho)(1-\chi)}{1-\rho\chi}$. Further, $Exec(s_2) = \{\emptyset\}$, hence, $\sum_{\Delta\in Exec(s_2)} PF(\Delta,s_2) = PF(\emptyset,s_2) = 1$ and $PT(\emptyset,s_2) = PM(s_2,s_2) = \frac{1}{1} = 1$. Finally, $PM(s_1,s_2) = PT(\{(\{a\},\rho)\},s_1) + PT(\{(\{a\},\chi)\},s_1) = \frac{\rho(1-\chi)}{1-\rho\chi} + \frac{\chi(1-\rho)}{1-\rho\chi} = \frac{\rho+\chi-2\rho\chi}{1-\rho\chi}$. In Table 6, the calculation of the probability functions $PF(\Gamma,s_1)$, $PT(\Gamma,s_1)$, $PM(s_1,s)$ is explained, where $\Gamma\in Exec(s_1)$, $s\in\{s_1,s_2\}$ (the value of s is depicted in the parentheses near the value of $PM(s_1,s)$) and $\Sigma = \sum_{\Delta\in Exec(s_1)} PX(\Delta,s_1)$, $PX\in\{PF,PT,PM\}$.*

Definition 3.5 *Let G be a dynamic expression. The* (labeled probabilistic) transition system *of G is a quadruple $TS(G) = (S_G, L_G, \mathcal{T}_G, s_G)$, where*

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

The definition of $TS(G)$ is correct, i.e. for every state, the sum of the probabilities of all the transitions starting from it is 1. This is guaranteed by the note after the definition of $PT(\Gamma, s)$. Thus, we have defined a *generative* model of probabilistic processes [71], according to the classification from [59]. The reason is that the sum of the probabilities of the transitions with all possible labels should be equal to 1, not only of those with the same labels (up to enumeration of activities they include) as in the *reactive* models [87, 88], and we do not have the nested probabilistic choice as in the *stratified* models [59].

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

Note that Γ can be the empty multiset, and its execution does not change the current state (i.e. the equivalence class), since we have a loop transition $s \xrightarrow{\emptyset}_{\mathcal{P}} s$ from a state s to itself as a result of executing the empty multiset. This corresponds to application of the empty loop rule to expressions from the equivalence class. We have to keep track of such executions, called *empty loops*, because they have nonzero probabilities. This follows from the definition of $PF(\emptyset, s)$ and the fact that multiaction probabilities cannot be equal to 1 as they belong to the interval $(0; 1)$.

The step probabilities belong to the interval $(0; 1]$. The step probability is 1 in the case when we cannot leave a state s, hence, the only transition leaving it is the empty loop transition $s \xrightarrow{\emptyset}_1 s$.

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

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

Definition 3.6 *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 $\beta : S_G \to S_{G'}$ 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. *$\forall s, \tilde{s} \in S_G\ \forall \Gamma\ s \xrightarrow{\Gamma}_{\mathcal{P}} \tilde{s} \ \Leftrightarrow\ \beta(s) \xrightarrow{\Gamma}_{\mathcal{P}} \beta(\tilde{s})$.*

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

Transition systems of static expressions can be defined as well. For $E \in \mathit{RegStatExpr}$, let $TS(E) = TS(\overline{E})$.

Definition 3.7 *Two dynamic expressions G and G' are* equivalent with respect to transition systems*, denoted by $G =_{ts} G'$, if $TS(G) \simeq TS(G')$.*

For a dynamic expression G, a discrete random variable is associated with every state $s \in DR(G)$. The variable captures a residence time in the state. One can interpret staying in a state at the next discrete time moment as a failure and leaving it as a success of some trial series. It is easy to see that the random variables are geometrically distributed with the parameter $1 - PM(s, s)$, since the probability to stay in the state $s \in DR(G)$ for $k - 1$ time moments and leave it at the moment $k \geq 1$ is $PM(s, s)^{k-1}(1 - PM(s, s))$ (the residence time is k in this case, and this formula defines the probability mass function (PMF) of residence time in s). Hence, the probability distribution function (PDF) of residence time in s is $1 - PM(s, s)^{k-1}$ $(k \geq 1)$ (the probability that the residence time in s is less than k). The mean value formula for the geometrical distribution allows us to calculate the *average sojourn time in the state s* as

$$SJ(s) = \frac{1}{1 - PM(s, s)}.$$

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

Analogously, the *sojourn time variance in the state s* is

$$VAR(s) = \frac{PM(s, s)}{(1 - PM(s, s))^2}.$$

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

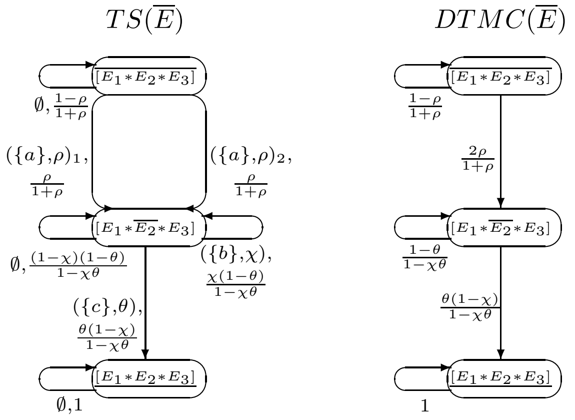

Figure 2: The transition system and the underlying DTMC of $\overline{E}$ for $E = [((\{a\},\rho)_1[](\{a\},\rho)_2)*(\{b\},\chi)*(\{c\},\theta)]$

Definition 3.8 *Let G be a dynamic expression. The* underlying discrete time Markov chain (DTMC) *of G, denoted by $DTMC(G)$, has the state space $DR(G)$, the initial state $[G]_\approx$ and the transitions $s \rightarrow_{\mathcal{P}} \tilde{s}$, if $s \rightarrow \tilde{s}$ and $\mathcal{P} = PM(s, \tilde{s})$.*

Underlying DTMCs of static expressions can be defined as well. For $E \in RegStatExpr$, let $DTMC(E) = DTMC(\overline{E})$.

Example 3.3 *Let $E_1 = (\{a\},\rho)[](\{a\},\rho)$, $E_2 = (\{b\},\chi)$, $E_3 = (\{c\},\theta)$ and $E = [E_1 * E_2 * E_3]$. The identical activities of the composite static expression are enumerated as follows: $E = [((\{a\},\rho)_1[](\{a\},\rho)_2) * (\{b\},\chi) * (\{c\},\theta)]$. In Figure 2, the transition system $TS(\overline{E})$ and the underlying DTMC $DTMC(\overline{E})$ are presented. For simplicity, the states are labeled by expressions belonging to the corresponding equivalence classes, and singleton multisets of activities are written without outer braces.*

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

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

Let us demonstrate how the transition probabilities are calculated. For instance, we have $PF(\{(\{a\},\rho)_1\}, s_1) = PF(\{(\{a\},\rho)_2\}, s_1) = \rho(1-\rho)$ and $PF(\emptyset, s_1) = (1-\rho)^2$. Hence, $\sum_{\Delta \in Exec(s_1)} PF(\Delta, s_1) = 2\rho(1-\rho) + (1-\rho)^2 = 1-\rho^2$. Thus, $PT(\{(\{a\},\rho)_1\}, s_1) = PT(\{(\{a\},\rho)_2\}, s_1) = \frac{\rho(1-\rho)}{1-\rho^2} = \frac{\rho(1-\rho)}{(1-\rho)(1+\rho)} = \frac{\rho}{1+\rho}$ and $PT(\emptyset, s_1) = \frac{(1-\rho)^2}{1-\rho^2} = \frac{(1-\rho)^2}{(1-\rho)(1+\rho)} = \frac{1-\rho}{1+\rho}$. The other probabilities are calculated in a similar way.

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

$$SJ = \left(\frac{1+\rho}{2\rho}, \frac{1-\chi\theta}{\theta(1-\chi)}, \infty\right).$$

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

$$VAR = \left(\frac{1-\rho^2}{4\rho^2}, \frac{(1-\theta)(1-\chi\theta)}{\theta^2(1-\chi)^2}, \infty\right).$$

4 Denotational semantics

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

4.1 Labeled DTSPNs

Let us introduce a class of labeled discrete time stochastic Petri nets (LDTSPNs), which are essentially a subclass of DTSPNs [99] (since we do not allow the transition probabilities to be equal to 1) extended with transition labeling. LDTSPNs are somewhat similar to labeled weighted DTSPNs (LWDTSPNs) from [36, 37], but in LWDTSPNs all transitions have weights, the transition probabilities may be equal to 1 and only maximal fireable subsets of the enabled transitions are fired.

First, we present a formal definition (construction, syntax) of LDTSPNs.

Definition 4.1 *A* labeled DTSPN (LDTSPN) *is a tuple* $N = (P_N, T_N, W_N, \Omega_N, \mathcal{L}_N, M_N)$*, where*

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

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

We now define a behaviour (functioning, semantics) of LDTSPNs.

Let N be an LDTSPN and $t \in T_N$, $U \in I\!N_{fin}^{T_N}$. The *precondition* ${}^\bullet t$ and the *postcondition* $t^\bullet$ of t are the multisets of places defined as $({}^\bullet t)(p) = W_N(p, t)$ and $(t^\bullet)(p) = W_N(t, p)$. The *precondition* ${}^\bullet U$ and the *postcondition* $U^\bullet$ of U are the multisets of places defined as ${}^\bullet U = \sum_{t \in U} {}^\bullet t$ and $U^\bullet = \sum_{t \in U} t^\bullet$. Note that for $U = \emptyset$ we have ${}^\bullet\emptyset = \emptyset = \emptyset^\bullet$.

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

Let M be a marking of an LDTSPN N. A transition $t \in Ena(M)$ fires with probability $\Omega_N(t)$ when no different transition is enabled, i.e. $Ena(M) = \{t\}$.

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

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

In the case $U = \emptyset$ we define

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

Thus, if $Ena(M) \neq \emptyset$ then $PF(U, M)$ can be interpreted as a *joint* probability of independent events (in a probability sense, i.e. the probability of intersection of these events is equal to the product of their probabilities). Each such an event consists in the positive or negative decision to fire of a particular transition. Every enabled transition decides probabilistically (using its probability) and independently (from others), if it wants to fire in M. If U is a set of all enabled transitions which have decided to fire in M and ${}^\bullet U \subseteq M$ then U is ready for firing in M. The multiplication in the definition is used because it reflects the probability of the independent event intersection. Alternatively, when $U \neq \emptyset$, $PF(U, M)$ can be interpreted as the probability that the set of transitions U fires *exclusively* in M, i.e. a the probability of *intersection* of two events calculated using the conditional probability formula in the form $\mathsf{P}(X \cap Y) = \mathsf{P}(X|Y)\mathsf{P}(Y)$. The event X consists in firing U

in M. The event Y consists in non-firing in M all the enabled transitions not belonging to U. Since the mentioned non-firings are obviously independent events, the probability of Y is a product of probabilities of the non-firings: $\mathsf{P}(Y) = \prod_{u\in Ena(M)\setminus U}(1-\Omega_N(u))$. The conditioning of X by Y makes the firings of the transitions from U independent, since all of them can fire in parallel in M due to the requirement ${}^\bullet U \subseteq M$. Hence, the probability that U fires *under condition* that no enabled transitions not belonging to U fire in M is a product of probabilities of these transitions: $\mathsf{P}(X|Y) = \prod_{t\in U}\Omega_N(t)$. Thus, the probability that U fires *and* no enabled transitions not belonging to U fire in M is the probability of X conditioned by Y multiplied by the probability of Y: $\mathsf{P}(X\cap Y) = \mathsf{P}(X|Y)\mathsf{P}(Y) = \prod_{t\in U}\Omega_N(t)\cdot\prod_{u\in Ena(M)\setminus U}(1-\Omega_N(u))$. When $U=\emptyset$, $PF(U,M)$ can be interpreted as the probability that no enabled transitions fire in M, thus, $PF(\emptyset,M) = \prod_{u\in Ena(M)}(1-\Omega_N(u))$. When no transitions are enabled in M, i.e. $Ena(M)=\emptyset$, we have $PF(\emptyset,M)=1$, since we stay in M in this case.

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

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

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

Thus, $PT(U,M)$ can be interpreted as the *conditional* probability that U fires in M calculated using the conditional probability formula in the form $\mathsf{P}(Z|W) = \frac{\mathsf{P}(Z\cap W)}{\mathsf{P}(W)}$. The event Z consists in the exclusive firing of U in M, hence, $\mathsf{P}(Z) = PF(U,M)$. The event W consists in the exclusive firing of any set (including the empty one) $V\subseteq Ena(M)$ in M such that ${}^\bullet V\subseteq M$. Thus, $W = \cup_j Z_j$, where $\forall j,\ Z_j$ are mutually exclusive events and $\exists i,\ Z = Z_i$ (in a probability sense, i.e. intersection of these events is the empty event). We have $\mathsf{P}(W) = \sum_j \mathsf{P}(Z_j) = \sum_{\{V\subseteq Ena(M)|{}^\bullet V\subseteq M\}} PF(V,M)$, because summation reflects the probability of the mutually exclusive event union. Since $Z\cap W = Z_i\cap(\cup_j Z_j) = Z_i = Z$, we have $\mathsf{P}(Z|W) = \frac{\mathsf{P}(Z)}{\mathsf{P}(W)} = \frac{PF(U,M)}{\sum_{\{V\subseteq Ena(M)|{}^\bullet V\subseteq M\}} PF(V,M)}$. $PF(U,M)$ can also be seen as the *potential* probability that U fires in M, since we have $PF(U,M) = PT(U,M)$ only when *all* subsets (including the empty one) of transitions from $Ena(M)$ can fire in M. In this case, all the mentioned transitions can fire in parallel in M (i.e. $Ena(M)$ can fire in M) and we have $\sum_{\{V\subseteq Ena(M)|{}^\bullet V\subseteq M\}} PF(V,M) = 1$, since this sum collects the products of *all* combinations of the transition probabilities and their negations. But in general, for example, for two transitions t and u enabled in M, it can happen that they cannot fire in M together, in parallel, i.e. $t,u\in Ena(M)$, but $\{t,u\}\not\in Ena(M)$. Note that $PT(\emptyset,M)\in(0;1]$, hence, there is a non-zero probability to stay in the marking M at the next time moment, and the residence time in M is at least 1 discrete time unit.

Let $Ena(M) = \{t_1,\ldots,t_n\}$ be a mutually exclusive set of transitions (i.e. firing of any transition from the set results in a marking in which no other transition from the set is enabled) and $\rho_i = \Omega_N(t_i)$ $(1\le i\le n)$. Then $PT(\{t_i\},M)$ resembles the probabilistic function $P[E_i]$ from [99], which defines the probability of the event E_i, that transition t_i in a mutually exclusive set of transitions $\{t_1,\ldots,t_n\}$ will fire in the marking M. We have $P[E_i] = \frac{\frac{\rho_i}{1-\rho_i}}{1+\sum_{j=1}^n\frac{\rho_j}{1-\rho_j}} = \frac{\frac{\rho_i(1-\rho_1)\cdots(1-\rho_n)}{1-\rho_i}}{(1-\rho_1)\cdots(1-\rho_n)+\sum_{j=1}^n\frac{\rho_j(1-\rho_1)\cdots(1-\rho_n)}{1-\rho_j}}$, where $\frac{\rho_i(1-\rho_1)\cdots(1-\rho_n)}{1-\rho_i}$ corresponds to $PF(\{t_i\},M)$ in our setting. Further, $PT(\emptyset,M)$ resembles the probabilistic function $P[E_0]$, which defines the probability of the event E_0, that no transitions from the mutually exclusive set of transitions $\{t_1,\ldots,t_n\}$ will fire in the marking M. We have $P[E_0] = \frac{1}{1+\sum_{j=1}^n\frac{\rho_j}{1-\rho_j}} = \frac{(1-\rho_1)\cdots(1-\rho_n)}{(1-\rho_1)\cdots(1-\rho_n)+\sum_{j=1}^n\frac{\rho_j(1-\rho_1)\cdots(1-\rho_n)}{1-\rho_j}}$, where $(1-\rho_1)\cdots(1-\rho_n)$ corresponds to $PF(\emptyset,M)$ in our setting. If $Ena(M)$ is not a mutually exclusive set of transitions, our way to define $PT(U,M)$ for $U\subseteq Ena(M)$, $U\neq\emptyset$, also extends the approach of [97,99]. The advantage of our two-stage definition of $PT(U,M)$ is that it has a closed form and we do not need to consider which sets of transitions are exclusive, instead, we just consider the probability that U fires in M under condition that only particular subsets of $Ena(M)$ can fire in M.

Note that for all markings of an LDTSPN N the sum of outgoing probabilities is equal to 1. More formally, $\forall M\in I\!N_{fin}^{P_N}\ \sum_{\{U\subseteq Ena(M)|{}^\bullet U\subseteq M\}} PT(U,M) = 1$. This obviously follows from the definition of $PT(U,M)$ and guarantees that it defines a probability distribution.

We write $M\stackrel{U}{\rightarrow}\widetilde{M}$ if $\exists\mathcal{P}\ M\stackrel{U}{\rightarrow}_{\mathcal{P}}\widetilde{M}$ and $M\rightarrow\widetilde{M}$ if $\exists U\ M\stackrel{U}{\rightarrow}\widetilde{M}$. For one-element set of transitions $U=\{t\}$, we write $M\stackrel{t}{\rightarrow}_{\mathcal{P}}\widetilde{M}$ and $M\stackrel{t}{\rightarrow}\widetilde{M}$.

Definition 4.2 *Let N be an LDTSPN. The* reachability set *of N, denoted by $RS(N)$, is the minimal set of markings such that*

- $M_N \in RS(N)$;
- *if* $M \in RS(N)$ *and* $M \to \widetilde{M}$ *then* $\widetilde{M} \in RS(N)$.

Definition 4.3 *Let N be an LDTSPN. The* reachability graph *of N is a (labeled probabilistic) transition system $RG(N) = (S_N, L_N, \mathcal{T}_N, s_N)$, where*

- *the set of* states *is* $S_N = RS(N)$;
- *the set of* labels *is* $L_N = 2^{T_N} \times (0;1]$;
- *the set of* transitions *is* $\mathcal{T}_N = \{(M, (U, \mathcal{P}), \widetilde{M}) \mid M, \widetilde{M} \in RS(N),\ M \xrightarrow{U}_{\mathcal{P}} \widetilde{M}\}$;
- *the* initial state *is* $s_N = M_N$.

Definition 4.4 *Let N be an LDTSPN. The* underlying discrete time Markov chain (DTMC) *of N, denoted by $DTMC(N)$, has the state space $RS(N)$, the initial state M_N and the transitions $M \to_{\mathcal{P}} \widetilde{M}$, if $M \to \widetilde{M}$, where $\mathcal{P} = PM(M, \widetilde{M})$ is the* probability to move from M to $\widetilde{M}$ by firing any set of transitions *defined as*

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

Since $PM(M, \widetilde{M})$ is the probability for *any* (including the empty one) transition set to change marking M to $\widetilde{M}$, we use summation in the definition. Note that $\forall M \in RS(N)\ \sum_{\{\widetilde{M} \mid M \to \widetilde{M}\}} PM(M, \widetilde{M}) = \sum_{\{\widetilde{M} \mid M \to \widetilde{M}\}} \sum_{\{U \mid M \xrightarrow{U} \widetilde{M}\}} PT(U, M) = \sum_{\{U \subseteq Ena(M) \mid {}^\bullet U \subseteq M\}} PT(U, M) = 1$.

Let N be an LDTSPN and $M \in RS(N)$. The *average sojourn time in the marking M* is

$$SJ(M) = \frac{1}{1 - PM(M, M)}.$$

The *average sojourn time vector* of N, denoted by SJ, has the elements $SJ(M),\ M \in RS(N)$.

The *sojourn time variance in the marking M* is

$$VAR(M) = \frac{PM(M, M)}{(1 - PM(M, M))^2}.$$

The *sojourn time variance vector* of N, denoted by VAR, has the elements $VAR(M),\ M \in RS(N)$.

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

The average sojourn time vector of N is

$$SJ = \left(\frac{1}{\rho + \chi - \rho\chi}, \frac{1}{\chi}, \frac{1}{\rho}, \frac{1}{\theta}\right).$$

The sojourn time variance vector of N is

$$VAR = \left(\frac{1 - \rho - \chi + \rho\chi}{(\rho + \chi - \rho\chi)^2}, \frac{1 - \chi}{\chi^2}, \frac{1 - \rho}{\rho^2}, \frac{1 - \theta}{\theta^2}\right).$$

The elements $\mathcal{P}_{ij}$ $(1 \le i, j \le 4)$ of the (one-step) transition probability matrix (TPM) for $DTMC(N)$ are defined as

$$\mathcal{P}_{ij} = \begin{cases} PM(M_i, M_j), & M_i \to M_j; \\ 0, & \textit{otherwise}. \end{cases}$$

Thus, the TPM is

$$\mathbf{P} = \begin{pmatrix} (1-\rho)(1-\chi) & \rho(1-\chi) & \chi(1-\rho) & \rho\chi \\ 0 & 1-\chi & 0 & \chi \\ 0 & 0 & 1-\rho & \rho \\ \theta & 0 & 0 & 1-\theta \end{pmatrix}.$$

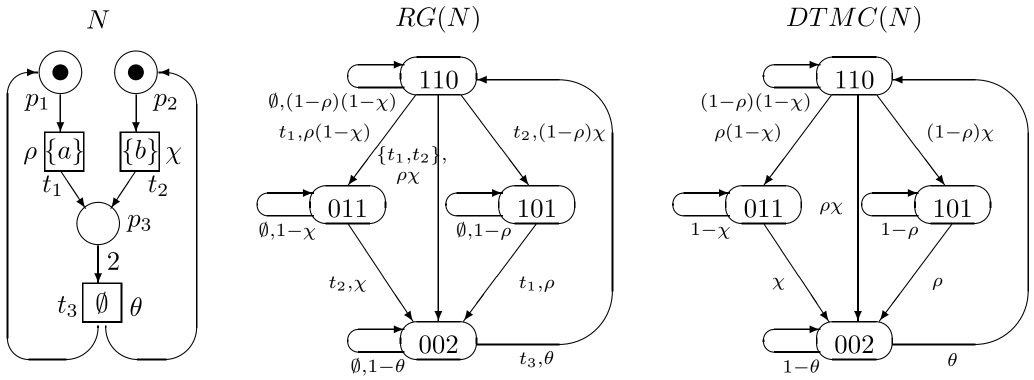

Figure 3: LDTSPN, its reachability graph and the underlying DTMC

The steady-state PMF $\psi = (\psi_1, \psi_2, \psi_3, \psi_4)$ *for* $DTMC(N)$ *is a solution of the equation system*

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

where $\mathbf{I}$ *is the identity matrix of size four and* $\mathbf{0} = (0, 0, 0, 0)$, $\mathbf{1} = (1, 1, 1, 1)$.
For the case $\rho = \chi = \theta$ *we have*

$$\psi = \left(\frac{1}{5 - 3\rho}, \frac{1 - \rho}{5 - 3\rho}, \frac{1 - \rho}{5 - 3\rho}, \frac{2 - \rho}{5 - 3\rho} \right).$$

The inverse of the steady-state PMF for $DTMC(N)$ *is its mean recurrence time vector*

$$RC = \left(5 - 3\rho, \frac{5 - 3\rho}{1 - \rho}, \frac{5 - 3\rho}{1 - \rho}, \frac{5 - 3\rho}{2 - \rho} \right).$$

Each element of RC *is the mean number of steps to return to the corresponding marking. For instance, one can see that the average time to come back to the initial marking* $M_N = M_1$ *in the long-term behaviour belongs in the interval* $(2; 5)$, *since* $\rho \in (0; 1)$.

4.2 Algebra of dts-boxes

We now propose discrete time stochastic Petri boxes and associated algebraic operations to define a net representation of *dtsPBC* expressions.

Definition 4.5 *A* discrete time stochastic Petri box (dts-box) *is a tuple* $N = (P_N, T_N, W_N, \Lambda_N)$, *where*

- P_N *and* T_N *are finite sets of* places *and* transitions, *respectively, such that* $P_N \cup T_N \neq \emptyset$ *and* $P_N \cap T_N = \emptyset$*;*
- $W_N : (P_N \times T_N) \cup (T_N \times P_N) \to I\!N$ *is a function providing the* weights of arcs *between places and transitions;*
- Λ_N *is the* place and transition labeling *function such that*
  - $\Lambda_N|_{P_N} : P_N \to \{\mathsf{e}, \mathsf{i}, \mathsf{x}\}$ *(it specifies* entry, internal *and* exit *places, respectively);*
  - $\Lambda_N|_{T_N} : T_N \to \{\varrho \mid \varrho \subseteq I\!N_{fin}^{\mathcal{SL}} \times \mathcal{SL}\}$ *(it associates transitions with the* relabeling relations *on activities).*

Moreover, $\forall t \in T_N \ {}^\bullet t \neq \emptyset \neq t^\bullet$. *In addition, for the set of* entry *places of* N *defined as* ${}^\circ N = \{p \in P_N \mid \Lambda_N(p) = \mathsf{e}\}$ *and the set of* exit *places of* N *defined as* $N^\circ = \{p \in P_N \mid \Lambda_N(p) = \mathsf{x}\}$, *the following condition holds:* ${}^\circ N \neq \emptyset \neq N^\circ$, ${}^\bullet({}^\circ N) = \emptyset = (N^\circ)^\bullet$.

A dts-box is *plain* if $\forall t \in T_N \ \exists (\alpha, \rho) \in \mathcal{SL} \ \Lambda_N(t) = \varrho_{(\alpha,\rho)}$, where $\varrho_{(\alpha,\rho)} = \{(\emptyset, (\alpha, \rho))\}$ is a constant relabeling that can be identified with the activity (α, ρ). A *marked plain dts-box* is a pair (N, M_N), where N is a plain dts-box and $M_N \in I\!N_{fin}^{P_N}$ is its marking. We shall use the following notation: $\overline{N} = (N, {}^\circ N)$ and $\underline{N} = (N, N^\circ)$. Note that a marked plain dts-box $(P_N, T_N, W_N, \Lambda_N, M_N)$ could be interpreted as the LDTSPN

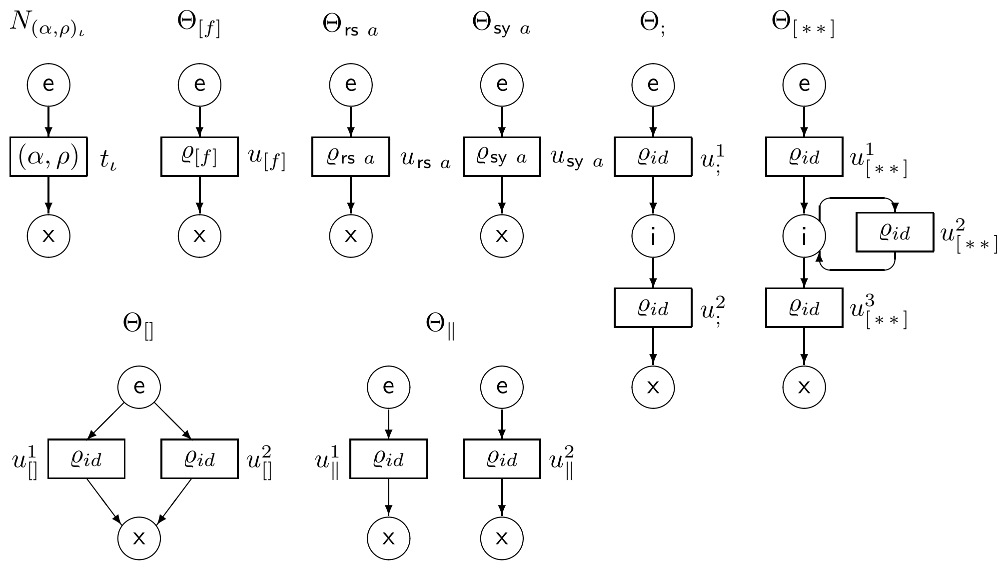

Figure 4: The plain and operator dts-boxes

$(P_N, T_N, W_N, \Omega_N, \mathcal{L}_N, M_N)$, where functions Ω_N and $\mathcal{L}_N$ are defined as follows: $\forall t \in T_N\ \Omega_N(t) = \rho$ and $\mathcal{L}_N(t) = \alpha$, where $\Lambda_N(t) = \varrho_{(\alpha,\rho)}$. Behaviour of the marked dts-boxes follows from the firing rule of LDTSPNs. A plain dts-box N is *n-bounded* ($n \in I\!N$) if $\overline{N}$ is so, i.e. $\forall M \in RS(\overline{N})\ \forall p \in P_N\ M(p) \leq n$, and it is *safe* if it is 1-bounded. A plain dts-box N is *clean* if $\forall M \in RS(\overline{N})\ {}^\circ N \subseteq M \ \Rightarrow\ M = {}^\circ N$ and $N^\circ \subseteq M \ \Rightarrow\ M = N^\circ$, i.e. if there are tokens in all its entry (exit) places then no other places have tokens.

The structure of the plain dts-box corresponding to a static expression is constructed like in *PBC* [15, 29, 30], i.e. we use a simultaneous refinement and relabeling meta-operator (net refinement) in addition to the *operator dts-boxes* corresponding to the algebraic operations of *dtsPBC* and featuring transformational transition relabelings. Operator dts-boxes specify n-ary functions from plain dts-boxes to plain dts-boxes (we have $1 \leq n \leq 3$ in *dtsPBC*). Thus, as we shall see in Theorem 4.1, the resulting plain dts-boxes are safe and clean. In the definition of the denotational semantics, we shall apply standard constructions used for *PBC*. Let Θ denote an *operator box* and u denote a *transition name* from the *PBC* setting.

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

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

The plain dts-box $N_{(\alpha,\rho)_\iota}$ and operator dts-boxes are presented in Figure 4. Note that the symbol i is usually omitted.

In the case of the iteration, a decision that we must take is the selection of the operator box that we shall use for it, since we have two proposals in plain *PBC* for that purpose [15]. One of them provides us with a safe version with six transitions in the operator box, but there is also a simpler version, which has only three transitions. In general, in PBC, with the latter version we may generate 2-bounded nets, which only occurs when a parallel behavior appears at the highest level of the body of the iteration. Nevertheless, in our case, and due to the syntactical restriction introduced for regular terms, this particular situation cannot occur, so that the net obtained will be always safe.

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

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

- $Box_{dts}((\alpha,\rho)_\iota) = N_{(\alpha,\rho)_\iota}$. Since a single transition t_ι corresponds to the activity $(\alpha,\rho)_\iota \in \mathcal{SL}$, their numberings coincide:

$$Enu(t_\iota) = \iota.$$

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

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

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

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

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

$$Enu(t) = Enu_E(t),\ t \in T_E,\ a, \hat{a} \notin \alpha,\ \Lambda_E(t) = \varrho_{(\alpha,\rho)}.$$

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

  Thus, the enumeration function is defined as

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

  When we synchronize the same set of transitions in different orders, we obtain several resulting transitions with the same label and probability, but with different numberings having the same content. In this case, we shall consider only a single transition from the resulting ones in the plain dts-box to avoid introducing redundant transitions. For example, if the transitions t and u are generated by synchronizing v and w in different orders, we have $\Lambda(t) = \varrho_{(\alpha \oplus_a \beta, \rho \cdot \chi)} = \Lambda(u)$, but $Enu(t) = (Enu_E(v))(Enu_E(w)) \neq (Enu_E(w))(Enu_E(v)) = Enu(u)$, whereas $Cont(Enu(t)) = Cont(Enu(v)) \cup Cont(Enu(w)) = Cont(Enu(u))$. Then only one transition t (or, symmetrically, u) will appear in $Box_{dts}(E \text{ sy } a)$.

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

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

We now can formally define the denotational semantics as a homomorphism.

Definition 4.6 *Let $(\alpha,\rho) \in \mathcal{SL},\ a \in Act$ and $E, F, K \in RegStatExpr$. The* denotational semantics *of $dtsPBC$ is a mapping Box_{dts} from $RegStatExpr$ into the area of plain dts-boxes defined as follows:*

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

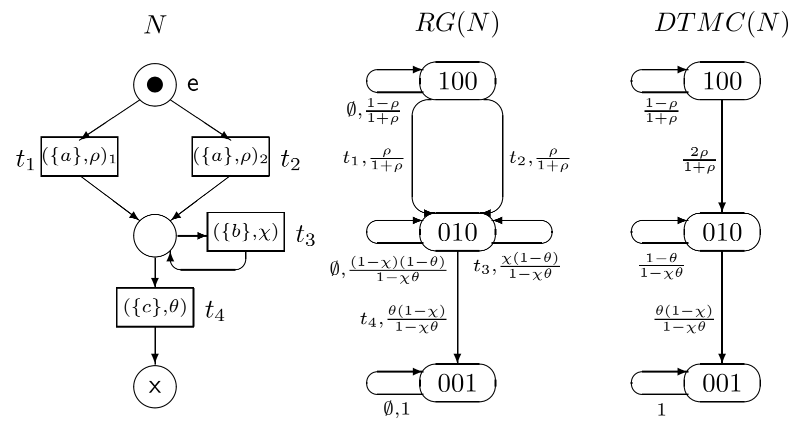

Figure 5: The marked dts-box $N = Box_{dts}(\overline{E})$ for $E = [((\{a\},\rho)_1[](\{a\},\rho)_2) * (\{b\},\chi) * (\{c\},\theta)]$, its reachability graph and the underlying DTMC

The dts-boxes of dynamic expressions can be defined as well. For $E \in RegStatExpr$, let $Box_{dts}(\overline{E}) = \overline{Box_{dts}(E)}$ and $Box_{dts}(\underline{E}) = \underline{Box_{dts}(E)}$.

Note that this definition is compositional in the sense that, for any arbitrary dynamic expression, we may decompose it in some inner dynamic and static expressions, for which we may apply the definition, thus obtaining the corresponding plain dts-boxes, which can be joined according to the term structure (by definition of Box_{dts}), the resulting plain box being marked in the places that were marked in the argument nets.

Theorem 4.1 *For any static expression E, $Box_{dts}(\overline{E})$ is safe and clean.*

Proof. The structure of the net is obtained as in PBC [15, 29, 30], combining both refinement and relabeling. Consequently, the dts-boxes thus obtained will be safe and clean. □

Let $\simeq$ denote the isomorphism between transition systems and reachability graphs or between DTMCs that binds their initial states. Due to the space restrictions, we omit the corresponding definitions as they resemble that of the isomorphism between transition systems. Note that the names of transitions of the dts-box corresponding to a static expression could be identified with the enumerated activities of the latter.

Theorem 4.2 *For any static expression E*

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

Proof. See Appendix A.1. □

Proposition 4.1 *For any static expression E*

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

Proof. By Theorem 4.2 and definitions of underlying DTMCs for dynamic expressions and LDTSPNs, since transition probabilities of the associated DTMCs are the sums of those belonging to transition systems or reachability graphs. □

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

Consider the next example that demonstrates synchronization.

Example 4.3 *Let $E_1 = (\{a\},\rho)$, $E_2 = (\{\hat{a}\},\chi)$ and $E = (E_1 \| E_2)$ sy $a = ((\{a\},\rho)\|(\{\hat{a}\},\chi))$ sy a. In Figure 6, the transition system $TS(\overline{E})$ and the underlying DTMC $DTMC(\overline{E})$ are presented. In Figure 7, the marked dts-box $N = Box_{dts}(\overline{E})$, its reachability graph $RG(N)$ and the underlying DTMC $DTMC(N)$ are depicted. It is easy to see that $TS(\overline{E})$ and $RG(N)$ are isomorphic, as well as $DTMC(\overline{E})$ and $DTMC(N)$.*

The probabilities $\mathcal{P}_{ij}$ $(1 \le i, j \le 4)$ are calculated as follows. Note that the symbol sy inscribes probability of the transition generated by synchronization, and the symbol $\|$ inscribes that of the transition corresponding

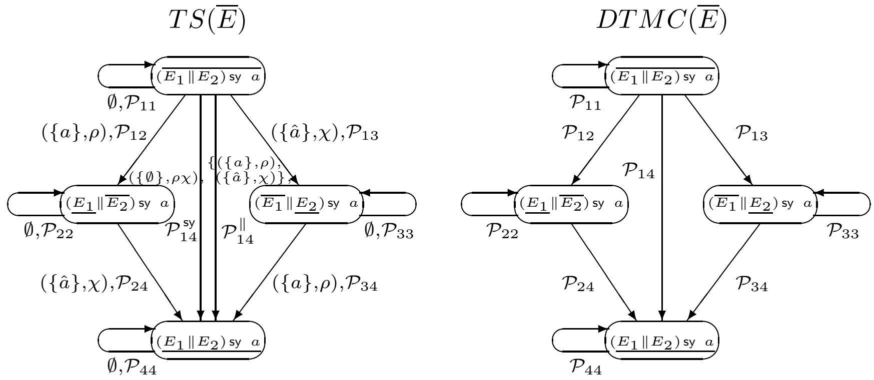

Figure 6: The transition system and the underlying DTMC of $\overline{E}$ for $E = ((\{a\},\rho)\|(\{\hat{a}\},\chi))$ sy a

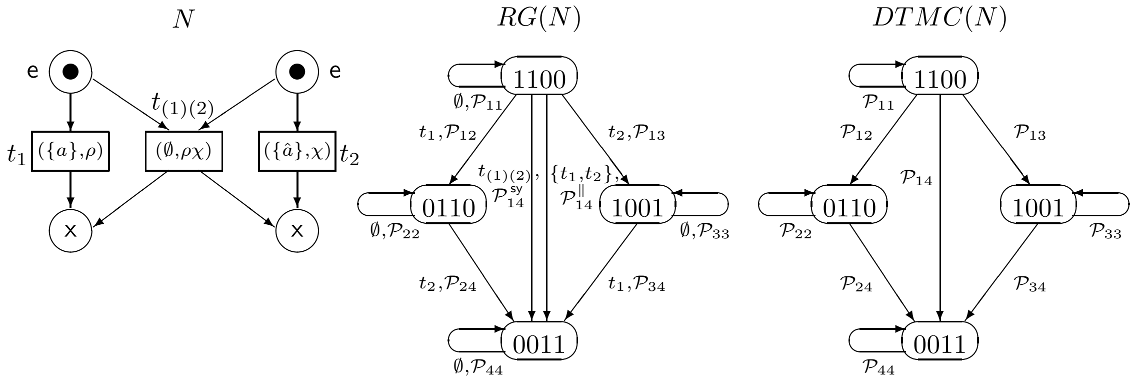

Figure 7: The marked dts-box $N = Box_{dts}(\overline{E})$ for $E = ((\{a\},\rho)\|(\{\hat{a}\},\chi))$ sy a, its reachability graph and the underlying DTMC

to the concurrent execution of two activities. To avoid complex notation, we use the normalization factor $\mathcal{N} = \frac{1}{1-\rho^2\chi-\rho\chi^2+\rho^2\chi^2}$.

$$\begin{array}{lll}
\mathcal{P}_{11} = \mathcal{N}(1-\rho)(1-\chi)(1-\rho\chi) & \mathcal{P}_{12} = \mathcal{N}\rho(1-\chi)(1-\rho\chi) & \mathcal{P}_{13} = \mathcal{N}\chi(1-\rho)(1-\rho\chi) \\
\mathcal{P}_{14}^{\mathsf{sy}} = \mathcal{N}\rho\chi(1-\rho)(1-\chi) & \mathcal{P}_{14}^{\|} = \mathcal{N}\rho\chi(1-\rho\chi) & \mathcal{P}_{22} = 1-\chi \\
\mathcal{P}_{24} = \chi & \mathcal{P}_{33} = 1-\rho & \mathcal{P}_{34} = \rho \\
\mathcal{P}_{44} = 1 & \mathcal{P}_{14} = \mathcal{P}_{14}^{\mathsf{sy}} + \mathcal{P}_{14}^{\|} = \mathcal{N}\rho\chi(2-\rho-\chi) &
\end{array}$$

Consider the case $\rho = \chi = \frac{1}{2}$. *Then the transition probabilities will be the following:*

$$\mathcal{P}_{11} = \mathcal{P}_{12} = \mathcal{P}_{13} = \mathcal{P}_{14}^{\|} = \frac{3}{13},\ \mathcal{P}_{14}^{\mathsf{sy}} = \frac{1}{13},\ \mathcal{P}_{22} = \mathcal{P}_{24} = \mathcal{P}_{33} = \mathcal{P}_{34} = \frac{1}{2},\ \mathcal{P}_{44} = 1,\ \mathcal{P}_{14} = \frac{4}{13}.$$

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

Example 4.4 *Let* $E = [((\{a\},\frac{1}{2}) * ((\{b\},\frac{1}{2})\|(\{c\},\frac{1}{2})) * (\{d\},\frac{1}{2})]$. *In Figure 8, the marked dts-box* $N = Box_{dts}(\overline{E})$ *and its reachability graph* $RG(N)$ *are presented. In the marking* $(0,1,1,2,0,0)$ *there are* 2 *tokens in the place* p_4. *Symmetrically, in the marking* $(0,1,1,0,2,0)$ *there are* 2 *tokens in the place* p_5. *Thus, allowing concurrency in the second argument of iteration in the expression* $\overline{E}$ *can lead to non-safeness of the corresponding marked dts-box* N*, though, it is* 2*-bounded in the worst case [15]. The origin of the problem is that* N *has as a self-loop with two subnets which can function independently. Therefore, we have decided to consider regular expressions only, since the alternative, which is a safe version of the iteration operator with six arguments in the corresponding dts-box, like that from [15], is rather cumbersome and has too intricate Petri net interpretation. Our motivation was to keep the algebraic and Petri net specifications as simple as possible.*

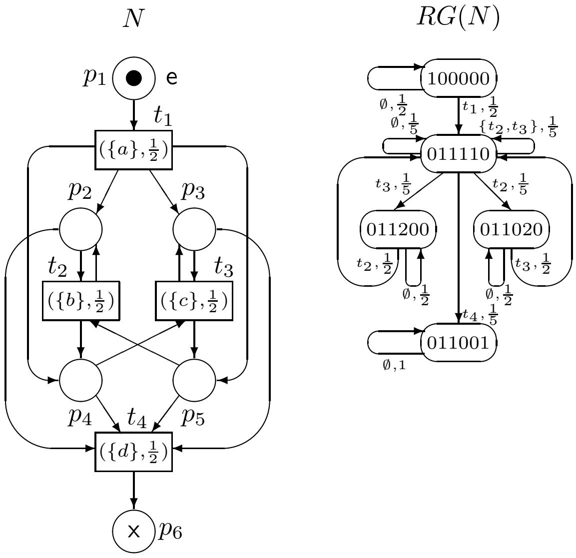

Figure 8: The marked dts-box $N = Box_{dts}(\overline{E})$ for $E = [((\{a\}, \frac{1}{2}) * ((\{b\}, \frac{1}{2}) \| (\{c\}, \frac{1}{2})) * (\{d\}, \frac{1}{2})]$ and its reachability graph

5 Stochastic equivalences

In this section, we propose a number of stochastic equivalences of expressions. The semantic equivalence $=_{ts}$ is too discriminating in many cases, i.e. from our viewpoint, it differentiates too many processes with similar behaviour. Hence, we need weaker equivalence notions to compare behaviour of processes specified by algebraic formulas.

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

The equivalences we intend to define 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.

To identify processes with intuitively similar behavior and to be able to apply standard constructions and techniques, we should abstract from infinite internal behaviour. Since *dtsPBC* is a stochastic extension of a finite part of *PBC* with iteration, the only source of infinite silent behaviour are empty loops, i.e. the transitions which are labeled by the empty multiset of activities and do not change states. During such an abstraction, we should collect the probabilities of empty loops. Note that the resulting probabilities are those defined for an infinite number of empty steps. In the following, we explain how to abstract from the empty loops both in the algebraic setting of *dtsPBC* and in the net one of LDTSPNs.

The result of abstraction from empty loops is a new semantics with the *progress* property: at least, one activity is executed at every discrete time tick. In this semantics, the sojourn time in a state can be greater than 1 only when the iteration body consisting of a single activity is executed. Notice that we do not consider as a silent behaviour the execution of an iteration body built only from activities with the empty multiaction parts, even when the body consists in a single activity $(\emptyset, \rho)$ whose execution does not change the current state of the transition system. The reason is that we skip only the empty steps at the considered abstraction level, but the iteration body consists in at least from one activity in this case.

5.1 Empty loops in transition systems

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

Let G be a dynamic expression and $s \in DR(G)$.

The *probability to stay in s due to k ($k \geq 1$) empty loops* is

$$(PT(\emptyset, s))^k.$$

Let $\Gamma \in Exec(s) \setminus \{\emptyset\}$, i.e. $PT(\emptyset, s) < 1$. The *probability to execute the non-empty multiset of activities Γ in s after possible empty loops* is

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

where $EL(s) = \frac{1}{1-PT(\emptyset,s)}$ is the *empty loops abstraction factor*. The *empty loops abstraction vector* of G, denoted by EL, has the elements $EL(s),\ s \in DR(G)$. The value $k = 0$ in the summation above corresponds to the case when no empty loops occur. $PT^*(\Gamma, s)$ can be interpreted as a conditional probability, under the condition that empty loops are left finally.

Notice that after abstraction from transition probabilities with empty multisets of activities, the remaining transition probabilities are normalized. In order to calculate transition probabilities $PT(\Gamma, s)$, we had to normalize $PF(\Gamma, s)$. Then, to obtain probabilities of non-empty steps $PT^*(\Gamma, s)$, we have to normalize $PT(\Gamma, s)$. Thus, we have a two-stage normalization as a result.

Note that $PT^*(\Gamma, s)$ defines a probability distribution, i.e. $\forall s \in DR(G)$ such that $PT(\emptyset, s) < 1$, i.e. there are non-empty steps after possible empty loops from s, we have $\sum_{\Gamma \in Exec(s)\setminus\{\emptyset\}} PT^*(\Gamma, s) = \frac{\sum_{\Gamma \in Exec(s)\setminus\{\emptyset\}} PT(\Gamma,s)}{1-PT(\emptyset,s)} = 1$, since $PT(\emptyset, s) + \sum_{\Gamma \in Exec(s)\setminus\{\emptyset\}} PT(\Gamma, s) = \sum_{\Delta \in Exec(s)} PT(\Delta, s) = 1$ and, hence, $\sum_{\Gamma \in Exec(s)\setminus\{\emptyset\}} PT(\Gamma, s) = 1 - PT(\emptyset, s)$.

Definition 5.1 *The* (labeled probabilistic) transition system without empty loops $TS^*(G)$ *has the state space* $DR(G)$ *and the transitions* $s \overset{\Gamma}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s}$, *if* $s \xrightarrow{\Gamma} \tilde{s},\ \Gamma \neq \emptyset$ *and* $\mathcal{P} = PT^*(\Gamma, s)$.

The definition of $TS^*(G)$ is correct, i.e. for every state $s \in DR(G)$ such that $PT(\emptyset, s) < 1$, the sum of the probabilities of all the transitions starting from it is 1. This is guaranteed by the note after the definition of $PT^*(\Gamma, s)$. If $PT(\emptyset, s) = 1$ then the sum of the exit probabilities for s in $TS^*(G)$ is 0.

Note that $TS^*(G)$ describes the viewpoint of a person who observes steps only if they include non-empty multisets of activities.

We write $s \overset{\Gamma}{\twoheadrightarrow} \tilde{s}$ if $\exists \mathcal{P}\ s \overset{\Gamma}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s}$ and $s \twoheadrightarrow \tilde{s}$ if $\exists \Gamma\ s \overset{\Gamma}{\twoheadrightarrow} \tilde{s}$. For a one-element multiset of activities $\Gamma = \{(\alpha, \rho)\}$, we write $s \overset{(\alpha,\rho)}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s}$ and $s \overset{(\alpha,\rho)}{\twoheadrightarrow} \tilde{s}$.

We decided to consider empty loops followed only by a non-empty step just for convenience. Alternatively, we could take a non-empty step succeeded by empty loops or a non-empty step preceded and succeeded by empty loops. In all these three cases our sequence begins or/and ends with the loops which do not change states. At the same time, the overall probabilities of the evolutions can differ, since empty loops have positive probabilities. To avoid inconsistency of definitions and too complex description, we consider sequences ending with a non-empty step. It resembles in some sense a construction of branching bisimulation [58].

Transition systems without empty loops of static expressions can be defined as well. For $E \in RegStatExpr$, let $TS^*(E) = TS^*(\overline{E})$.

Definition 5.2 *Two dynamic expressions G and G' are* equivalent with respect to transition systems without empty loops, *denoted by* $G =_{ts*} G'$, *if* $TS^*(G) \simeq TS^*(G')$.

Definition 5.3 *The* underlying DTMC without empty loops $DTMC^*(G)$ *has the state space* $DR(G)$ *and the transitions* $s \twoheadrightarrow_{\mathcal{P}} \tilde{s}$, *if* $s \twoheadrightarrow \tilde{s}$, *where* $\mathcal{P} = PM^*(s, \tilde{s})$ *is the* probability to move from s to $\tilde{s}$ by executing any non-empty multiset of activities after possible empty loops *defined as*

$$PM^*(s, \tilde{s}) = \sum_{\{\Gamma \mid s \overset{\Gamma}{\twoheadrightarrow} \tilde{s}\}} PT^*(\Gamma, s) = \begin{cases} EL(s)(PM(s, s) - PT(\emptyset, s)), & s = \tilde{s}; \\ EL(s)PM(s, \tilde{s}), & \textit{otherwise}, \end{cases}$$

where $PM(s, s) - PT(\emptyset, s)$ *is the* probability to stay in s due to any non-empty loop, *i.e. by executing any non-empty multiset of activities.*

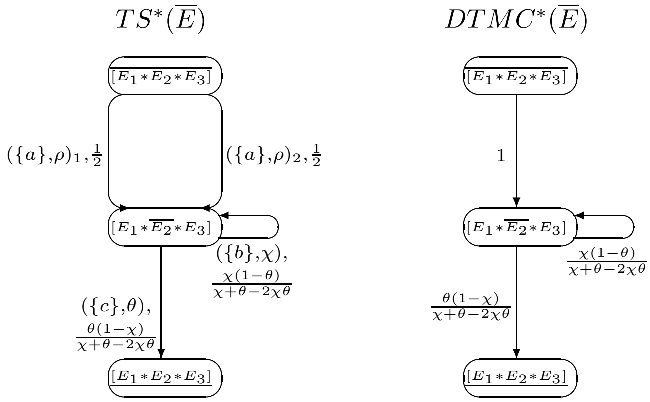

Figure 9: The transition system and the underlying DTMC without empty loops of $\overline{E}$ for $E = [((\{a\},\rho)_1[]$ $(\{a\},\rho)_2) * (\{b\},\chi) * (\{c\},\theta)]$

Note that $\forall s \in DR(G)$ such that $PT(\emptyset, s) < 1$, we have $\sum_{\{\tilde{s}|s \twoheadrightarrow \tilde{s}\}} PM^*(s,\tilde{s}) = \sum_{\{\tilde{s}|s \twoheadrightarrow \tilde{s}\}} \sum_{\{\Gamma|s \overset{\Gamma}{\twoheadrightarrow} \tilde{s}\}} PT^*(\Gamma, s) = \sum_{\Gamma \in Exec(s)\setminus\{\emptyset\}} PT^*(\Gamma, s) = 1$.

Underlying DTMCs without empty loops of static expressions can be defined as well. For $E \in RegStatExpr$, let $DTMC^*(E) = DTMC^*(\overline{E})$.

Example 5.1 *Let E be from Example 3.3. In Figure 9, the transition system $TS^*(\overline{E})$ and the underlying DTMC without empty loops $DTMC^*(\overline{E})$ are presented.*

Let us demonstrate how the transition probabilities of non-empty steps are calculated. For instance, we have $PT(\emptyset, s_1) = \frac{1-\rho}{1+\rho}$ and $\frac{1}{1-PT(\emptyset,s_1)} = \frac{1+\rho}{2\rho}$. Hence, since $PT(\{(\{a\},\rho)_1\}, s_1) = \frac{\rho}{1+\rho}$, we have $PT^*(\{(\{a\},\rho)_1\}, s_1) = \frac{PT(\{(\{a\},\rho)_1\},s_1)}{1-PT(\emptyset,s_1)} = \frac{\rho}{1+\rho} \cdot \frac{1+\rho}{2\rho} = \frac{1}{2}$. According to the same pattern, we obtain $PT^*(\{(\{a\},\rho)_2\}, s_1) = \frac{1}{2}$. The other probabilities are calculated in a similar way.

5.2 Empty loops in reachability graphs

Let N be an LDTSPN. Reachability graph $RG(N)$ can have loops going from a marking to itself which are labeled by the empty set and have non-zero probability. Such *empty loop* $M \overset{\emptyset}{\to}_{\mathcal{P}} M$ appears when no transitions fire at a discrete time tick, and this occurs with some positive probability. Obviously, in this case the current marking remains unchanged.

Let N be an LDTSPN and $M \in RS(N)$.

The *probability to stay in M due to k ($k \geq 1$) empty loops* is

$$(PT(\emptyset, M))^k.$$

Let $U \subseteq Ena(M)$, $U \neq \emptyset$ and ${}^\bullet U \subseteq M$, i.e. $PT(\emptyset, s) < 1$. The *probability that the non-empty set of transitions U fires in M after possible empty loops* is

$$PT^*(U, M) = PT(U, M) \sum_{k=0}^{\infty} (PT(\emptyset, M))^k = \frac{PT(U, M)}{1 - PT(\emptyset, M)} = EL(M)PT(U, M),$$

where $EL(M) = \frac{1}{1-PT(\emptyset,M)}$ is the *empty loops abstraction factor*. The *empty loops abstraction vector* of N, denoted by EL, has the elements $EL(M),\ M \in RS(N)$. The value $k = 0$ in the summation above corresponds to the case when no empty loops occur. $PT^*(U, M)$ can be interpreted as a conditional probability, under the condition that empty loops are left finally.

Notice that after abstraction from firing probabilities of empty sets of transitions, the remaining firing probabilities are normalized. In order to calculate firing probabilities $PT(U, M)$, we had to normalize $PF(U, M)$. Then, to obtain probabilities of non-empty steps $PT^*(U, M)$, we have to normalize $PT(U, M)$. Thus, we have a two-stage normalization as a result.

Note that $PT^*(U, M)$ defines a probability distribution, i.e. $\forall M \in RS(N)$ such that $PT(\emptyset, M) < 1$, i.e. there are non-empty steps after possible empty loops from M, we have $\sum_{\{U \neq \emptyset | {}^\bullet U \subseteq M\}} PT^*(U, M) =$

$\frac{\sum_{\{U\neq\emptyset|^\bullet U\subseteq M\}} PT(U,M)}{1-PT(\emptyset,M)} = 1$, since $PT(\emptyset,M) + \sum_{\{U\neq\emptyset|^\bullet U\subseteq M\}} PT(U,M) = \sum_{\{V\subseteq Ena(M)|^\bullet V\subseteq M\}} PT(V,M) = 1$ and, hence, $\sum_{\{U\neq\emptyset|^\bullet U\subseteq M\}} PT(U,M) = 1 - PT(\emptyset,M)$.

Definition 5.4 *The* reachability graph without empty loops $RG^*(N)$ *has the set of nodes* $RS(N)$ *and the arcs corresponding to the transitions* $M \stackrel{U}{\twoheadrightarrow}_{\mathcal{P}} \widetilde{M}$, *if* $M \stackrel{U}{\rightarrow} \widetilde{M}$, $U \neq \emptyset$ *and* $\mathcal{P} = PT^*(U,M)$.

Note that $RG^*(N)$ describes the viewpoint of a person who observes steps only if they include non-empty transition sets.

We write $M \stackrel{U}{\twoheadrightarrow} \widetilde{M}$ if $\exists\mathcal{P}\ M \stackrel{U}{\twoheadrightarrow}_{\mathcal{P}} \widetilde{M}$ and $M \twoheadrightarrow \widetilde{M}$ if $\exists U\ M \stackrel{U}{\twoheadrightarrow} \widetilde{M}$. For a one-element set of transitions $U = \{t\}$, we write $M \stackrel{t}{\twoheadrightarrow}_{\mathcal{P}} \widetilde{M}$ and $M \stackrel{t}{\twoheadrightarrow} \widetilde{M}$.

Definition 5.5 *The* underlying DTMC without empty loops $DTMC^*(N)$ *has the state space* $RS(N)$ *and the transitions* $M \twoheadrightarrow_{\mathcal{P}} \widetilde{M}$, *if* $M \twoheadrightarrow \widetilde{M}$, *where* $\mathcal{P} = PM^*(M,\widetilde{M})$ *is the* probability to move from M to $\widetilde{M}$ by firing any non-empty set of transitions after possible empty loops *defined as*

$$PM^*(M,\widetilde{M}) = \sum_{\{U|M\stackrel{U}{\twoheadrightarrow}\widetilde{M}\}} PT^*(U,M) = \begin{cases} EL(M)(PM(M,M) - PT(\emptyset,M)), & M = \widetilde{M};\\ EL(M)PM(M,\widetilde{M}), & \textit{otherwise}, \end{cases}$$

where $PM(M,M) - PT(\emptyset,M)$ *is the* probability to stay in M due to any non-empty loop, *i.e. by firing any non-empty multiset of transitions.*

Note that $\forall M \in RS(N)$ such that $PT(\emptyset,M) < 1$, we have $\sum_{\{\widetilde{M}|M\twoheadrightarrow\widetilde{M}\}} PM^*(M,\widetilde{M}) = \sum_{\{\widetilde{M}|M\twoheadrightarrow\widetilde{M}\}} \sum_{\{U|M\stackrel{U}{\twoheadrightarrow}\widetilde{M}\}} PT^*(U,M) = \sum_{\{U\neq\emptyset|^\bullet U\subseteq M\}} PT^*(U,M) = 1$.

Theorem 5.1 *For any static expression* E

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

Proof. As Theorem 4.2. □

Proposition 5.1 *For any static expression* E

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

Proof. As Proposition 4.1. □

Note that Theorem 5.1 guarantees that the net versions of algebraic equivalences could be easily defined. For every equivalence on the transition system without empty loops of a dynamic expression, a similarly defined analogue exists on the reachability graph without empty loops of the corresponding dts-box.

Example 5.2 *Let* E *be from Example 3.3 and* N *be from Example 4.2. In Figure 10, the reachability graph* $RG^*(N)$ *and the underlying DTMC without empty loops* $DTMC^*(N)$ *are presented. It is easy to see that* $TS^*(\overline{E})$ *and* $RG^*(N)$ *are isomorphic, as well as* $DTMC^*(\overline{E})$ *and* $DTMC^*(N)$.

Consider the next example that demonstrates synchronization.

Example 5.3 *Let* E *and* N *be those from Example 4.3. In Figure 11, the transition system* $TS^*(\overline{E})$ *and the underlying DTMC without empty loops* $DTMC^*(\overline{E})$ *are presented. In Figure 12, the reachability graph* $RG^*(N)$ *and the underlying DTMC without empty loops* $DTMC^*(N)$ *are depicted. It is easy to see that* $TS^*(\overline{E})$ *and* $RG^*(N)$ *are isomorphic, as well as* $DTMC^*(\overline{E})$ *and* $DTMC^*(N)$.

The probabilities $\mathcal{P}^*_{ij}$ $(1 \leq i,j \leq 4)$ *are calculated as follows. Note that the symbol* sy *inscribes probability of the transition generated by synchronization, and the symbol* $\|$ *inscribes that of the transition corresponding to the concurrent execution of two activities. To avoid complex notation, we use the normalization factor* $\mathcal{N}^* = \frac{1}{\rho+\chi-2\rho^2\chi-2\rho\chi^2+2\rho^2\chi^2}$. *The probabilities* $\mathcal{P}_{ij}$ $(1 \leq i,j \leq 4)$ *are taken from Example 4.3.*

$$\mathcal{P}^*_{12} = \frac{\mathcal{P}_{12}}{1-\mathcal{P}_{11}} = \mathcal{N}^*\rho(1-\chi)(1-\rho\chi) \qquad \mathcal{P}^*_{13} = \frac{\mathcal{P}_{13}}{1-\mathcal{P}_{11}} = \mathcal{N}^*\chi(1-\rho)(1-\rho\chi)$$
$$\mathcal{P}^{\mathsf{sy}*}_{14} = \frac{\mathcal{P}^{\mathsf{sy}}_{14}}{1-\mathcal{P}_{11}} = \mathcal{N}^*\rho\chi(1-\rho)(1-\chi) \qquad \mathcal{P}^{\|*}_{14} = \frac{\mathcal{P}^{\|}_{14}}{1-\mathcal{P}_{11}} = \mathcal{N}^*\rho\chi(1-\rho\chi)$$
$$\mathcal{P}^*_{24} = \frac{\mathcal{P}_{24}}{1-\mathcal{P}_{22}} = 1 \qquad \mathcal{P}^*_{34} = \frac{\mathcal{P}_{34}}{1-\mathcal{P}_{33}} = 1$$
$$\mathcal{P}^*_{14} = \mathcal{P}^{\mathsf{sy}*}_{14} + \mathcal{P}^{\|*}_{14} = \frac{\mathcal{P}^{\mathsf{sy}}_{14}+\mathcal{P}^{\|}_{14}}{1-\mathcal{P}_{11}} = \mathcal{N}^*\rho\chi(2-\rho-\chi)$$

Consider the case $\rho = \chi = \frac{1}{2}$. *Then the transition probabilities will be the following:*

$$\mathcal{P}^*_{12} = \mathcal{P}^*_{13} = \mathcal{P}^{\|*}_{14} = \frac{3}{10},\ \mathcal{P}^{\mathsf{sy}*}_{14} = \frac{1}{10},\ \mathcal{P}^*_{24} = \mathcal{P}^*_{34} = 1,\ \mathcal{P}^*_{14} = \frac{2}{5}.$$

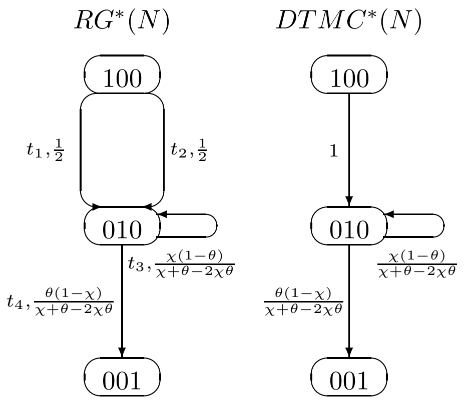

Figure 10: The reachability graph and the underlying DTMC without empty loops of $N = Box_{dts}(\overline{E})$ for $E = [((\{a\},\rho) * ((\{b\},\chi)\|(\{c\},\theta)) * (\{d\},\phi)]$

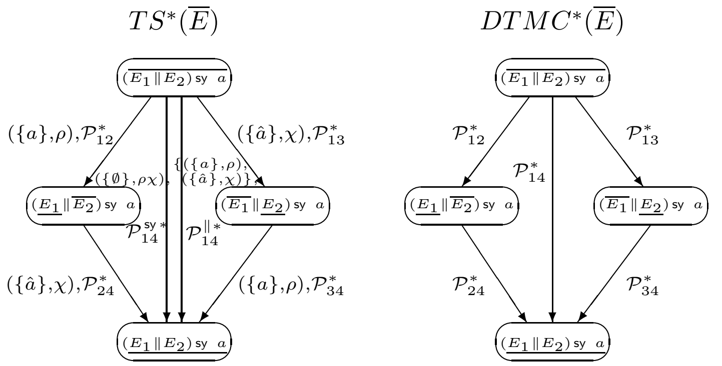

Figure 11: The transition system and the underlying DTMC without empty loops of $\overline{E}$ for $E = ((\{a\},\rho)\|$ $(\{\hat{a}\},\chi))$ sy a

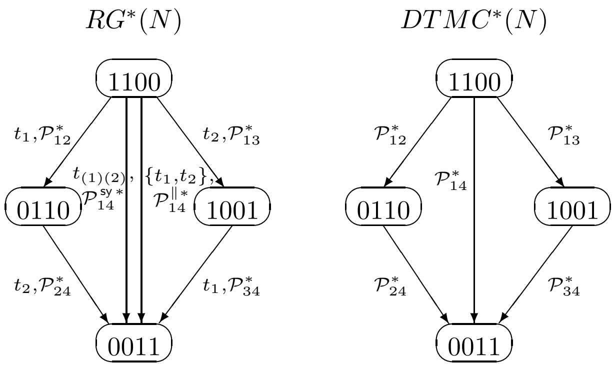

Figure 12: The reachability graph and the underlying DTMC without empty loops of $N = Box_{dts}(\overline{E})$ for $E = ((\{a\},\rho)\|(\{\hat{a}\},\chi))$ sy a

5.3 Stochastic trace equivalences

Trace equivalences are the least discriminating ones. In a trace semantics, the behavior of a system is associated with the set of all possible sequences of activities, i.e. protocols of work or computations. Thus, the points of choice of an external observer between several extensions of a particular computation are not taken into account.

Formal definitions of stochastic trace relations resemble those of trace equivalences for standard Petri nets [128] or process algebras, but additionally we have to take into account the probabilities of sequences of (multisets of) actions like in [43, 151]. First, we have to multiply occurrence probabilities for all (multisets of) activities along every path starting from the initial state of the transition system corresponding to a dynamic expression. The product is the probability of the sequence of multiaction parts of the (multisets of) activities along the path. Second, we should calculate a sum of probabilities for all paths corresponding to the same sequence of multiaction parts.

When concurrency aspects are not relevant, the interleaving behaviour is to be considered. The interleaving semantics abstracts from steps with more than one element. After such an abstraction, one has to normalize the probabilities of the remaining one-element steps. We need to do this since the sum of outgoing probabilities should always be equal to one for each state to form a probability distribution. For this, a special *interleaving transition relation* is proposed. Let G be a dynamic expression, $s, \tilde{s} \in DR(G)$ and $s \overset{(\alpha,\rho)}{\twoheadrightarrow} \tilde{s}$. We write $s \overset{(\alpha,\rho)}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s}$, where $\mathcal{P} = pt^*((\alpha, \rho), s)$ is the *probability to execute the activity* (α, ρ) *in* s *after possible empty loops, when only one-element steps are allowed*, defined as

$$pt^*((\alpha, \rho), s) = \frac{PT^*(\{(\alpha, \rho)\}, s)}{\sum_{\{(\beta,\chi)\}\in Exec(s)} PT^*(\{(\beta, \chi)\}, s)}.$$

Note that we have first abstracted from empty loops and then from steps with more than one element. We could perform the abstractions in the reverse order, the result will be the same. The reason is that, at every stage, we abstract from some transitions of a given transition system and then normalize the probabilities of the remaining ones. Hence, the result of each sequence of abstractions and normalizations coincides with that of the abstraction at once from all the transitions we have abstracted from in this sequence and the subsequent overall normalization. Let us prove it formally.

Let $\Gamma \in Exec(s)$ and $|\Gamma| \leq 1$. The *probability to execute the multiset of activities* Γ *in* s*, when only zero-element steps (i.e. empty loops) or one-element steps are allowed*, is

$$pt(\Gamma, s) = \frac{PT(\Gamma, s)}{PT(\emptyset, s) + \sum_{\{(\beta,\chi)\}\in Exec(s)} PT(\{(\beta, \chi)\}, s)}.$$

When we have first abstracted from empty loops and then from steps with more than one element, we get $pt^*((\alpha, \rho), s) = \frac{PT^*(\{(\alpha,\rho)\},s)}{\sum_{\{(\beta,\chi)\}\in Exec(s)} PT^*(\{(\beta,\chi)\},s)} = \frac{\frac{1}{1-PT(\emptyset,s)} PT(\{(\alpha,\rho)\},s)}{\frac{1}{1-PT(\emptyset,s)} \sum_{\{(\beta,\chi)\}\in Exec(s)} PT(\{(\beta,\chi)\},s)} = \frac{PT(\{(\alpha,\rho)\},s)}{\sum_{\{(\beta,\chi)\}\in Exec(s)} PT(\{(\beta,\chi)\},s)}$.

When we have first abstracted from steps with more than one element and then from empty loops, we get $pt^*((\alpha, \rho), s) = \frac{pt(\{(\alpha,\rho)\},s)}{\sum_{\{(\beta,\chi)\}\in Exec(s)} pt(\{(\beta,\chi)\},s)} = \frac{\frac{1}{PT(\emptyset,s)+\sum_{\{(\gamma,\theta)\}\in Exec(s)} PT(\{(\gamma,\theta)\},s)} PT(\{(\alpha,\rho)\},s)}{\frac{1}{PT(\emptyset,s)+\sum_{\{(\gamma,\theta)\}\in Exec(s)} PT(\{(\gamma,\theta)\},s)} \sum_{\{(\beta,\chi)\}\in Exec(s)} PT(\{(\beta,\chi)\},s)} = \frac{PT(\{(\alpha,\rho)\},s)}{\sum_{\{(\beta,\chi)\}\in Exec(s)} PT(\{(\beta,\chi)\},s)}$, i.e. the same expression as in the previous paragraph.

Note that in the definitions below we shall consider $\mathcal{L}(\Gamma) \in I\!N_{fin}^{\mathcal{L}}$ for $\Gamma \in I\!N_{fin}^{\mathcal{SL}}$, i.e. i.e. (possibly empty) multisets of multiactions. The multiactions can be empty as well. In this case, $\mathcal{L}(\Gamma)$ contains the elements $\emptyset$, but it is not empty itself.

Definition 5.6 *An* interleaving stochastic trace *of a dynamic expression* G *is a pair* $(\sigma, pt^*(\sigma))$*, where* $\sigma = \alpha_1 \cdots \alpha_n \in \mathcal{L}^*$ *and*

$$pt^*(\sigma) = \sum_{\{(\alpha_1,\rho_1),\dots,(\alpha_n,\rho_n) \mid [G]_\approx = s_0 \overset{(\alpha_1,\rho_1)}{\twoheadrightarrow} s_1 \overset{(\alpha_2,\rho_2)}{\twoheadrightarrow} \cdots \overset{(\alpha_n,\rho_n)}{\twoheadrightarrow} s_n\}} \prod_{i=1}^{n} pt^*((\alpha_i, \rho_i), s_{i-1}).$$

We denote the set of all interleaving stochastic traces *of a dynamic expression* G *by* $IntStochTraces(G)$*. Two dynamic expressions* G *and* G' *are* interleaving stochastic trace equivalent*, denoted by* $G \equiv_{is} G'$*, if*

$$IntStochTraces(G) = IntStochTraces(G').$$

Definition 5.7 *A* step stochastic trace *of a dynamic expression* G *is a pair* $(\Sigma, PT^*(\Sigma))$*, where* $\Sigma = A_1 \cdots A_n \in (I\!N_{fin}^{\mathcal{L}} \setminus \{\emptyset\})^*$ *and*

$$PT^*(\Sigma) = \sum_{\{\Gamma_1,\ldots,\Gamma_n \mid [G]_\approx = s_0 \stackrel{\Gamma_1}{\twoheadrightarrow} s_1 \stackrel{\Gamma_2}{\twoheadrightarrow} \cdots \stackrel{\Gamma_n}{\twoheadrightarrow} s_n,\ \mathcal{L}(\Gamma_i)=A_i\ (1\leq i\leq n)\}} \prod_{i=1}^{n} PT^*(\Gamma_i, s_{i-1}).$$

We denote the set of all step stochastic traces *of a dynamic expression G by $StepStochTraces(G)$. Two dynamic expressions G and G' are* step stochastic trace equivalent*, denoted by $G \equiv_{ss} G'$, if*

$$StepStochTraces(G) = StepStochTraces(G').$$

Example 5.4 *Let $E = ((\{a\}, \frac{1}{2})\|(\{\hat{a}\}, \frac{1}{2}))$ sy a. Then $IntStochTraces(\overline{E}) = \{(\emptyset, \frac{1}{7}),\ (\{a\}, \frac{3}{7}),\ (\{\hat{a}\}, \frac{3}{7}),$ $(\{a\}\{\hat{a}\}, \frac{3}{7}),\ (\{\hat{a}\}\{a\}, \frac{3}{7})\}$ and $StepStochTraces(\overline{E}) = \{(\{\emptyset\}, \frac{1}{10}),\ (\{\{a\}\}, \frac{3}{10}),\ (\{\{\hat{a}\}\}, \frac{3}{10}),\ (\{\{a\}\}\{\{\hat{a}\}\}, \frac{3}{10}),$ $(\{\{\hat{a}\}\}\{\{a\}\}, \frac{3}{10}),\ (\{\{\hat{a}\}, \{a\}\}, \frac{3}{10})\}$.*

5.4 Stochastic bisimulation equivalences

Bisimulation equivalences respect the particular points of choice in the behavior of a modeled system. We intend to present a definition of stochastic bisimulation equivalences. The definition is parameterized for the cases of interleaving or step semantics.

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

Thus, in our definitions, we follow the approach of [21, 23, 64, 70, 71, 87, 88], 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 [71], in contrast to the reactive model, treated in [87, 88], and we take transition probabilities instead of transition rates from [21, 23, 64, 70]. Thus, stochastic bisimulation equivalences, which we define further are (in the probabilistic sense) comparable only with interleaving probabilistic bisimulation equivalence from [71], and our step equivalence is obviously stronger while our interleaving one is similar to the mentioned relation.

Hence, the difference between bisimulation and trace equivalences is that we do not consider *all possible* occurrences of (multisets of) multiactions from the initial states, but only such that lead (stepwise) to the states *belonging to the same equivalence class*. Note that our interleaving stochastic bisimulation equivalence resembles in some sense weak bisimulation one from [24, 25], but we abstract from empty loops only instead of any transitions with the initial and the final states from the same equivalence class (with respect to the mentioned equivalence).

First, we introduce several helpful notations. Let G be a dynamic expression and $\mathcal{H} \subseteq DR(G)$. Then, for any $s \in DR(G)$ and $A \in I\!N_{fin}^{\mathcal{L}} \setminus \{\emptyset\}$, we write $s \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}} \mathcal{H}$, where $\mathcal{P} = PM_A^*(s, \mathcal{H})$ is the *overall probability to move from s into the set of states $\mathcal{H}$ via non-empty steps with the multiaction part A after possible empty loops* defined as

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

The summation in the definition above reflects the probability of the event union.

We write $s \stackrel{A}{\twoheadrightarrow} \mathcal{H}$ if $\exists \mathcal{P}\ s \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}} \mathcal{H}$.

We write $s \twoheadrightarrow_{\mathcal{P}} \mathcal{H}$ if $\exists A\ s \stackrel{A}{\twoheadrightarrow} \mathcal{H}$, where $\mathcal{P} = PM^*(s, \mathcal{H})$ is the *overall probability to move from s into the set of states $\mathcal{H}$ via any non-empty steps after possible empty loops* defined as

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

We propose the corresponding *interleaving transition relation $s \stackrel{\alpha}{\rightharpoondown}_{\mathcal{P}} \mathcal{H}$*, where $\mathcal{P} = pm_\alpha^*(s, \mathcal{H})$ is the *overall probability to move from s into the set of states $\mathcal{H}$ via steps with the multiaction part $\{\alpha\}$ after possible empty loops, when only one-element steps are allowed*, defined as

$$pm_\alpha^*(s,\mathcal{H}) = \sum_{\{(\alpha,\rho)\mid \exists \tilde{s}\in\mathcal{H}\ s \stackrel{(\alpha,\rho)}{\to} \tilde{s}\}} pt^*((\alpha,\rho),s).$$

We write $s \stackrel{\alpha}{\rightharpoonup} \mathcal{H}$ if $\exists \mathcal{P}\ s \stackrel{\alpha}{\rightharpoonup}_{\mathcal{P}} \mathcal{H}$.

To introduce a stochastic bisimulation between dynamic expressions G and G', we should consider the "composite" set of states $DR(G) \cup DR(G')$. The reason is that we have to identify the probabilities to come from any two equivalent states into the same "composite" equivalence class (with respect to the stochastic bisimulation) on this set. Note that, for $G \neq G'$, transitions starting from the states of $DR(G)$ (or $DR(G')$) always lead to those from the same set, since $DR(G) \cap DR(G') = \emptyset$, and this allows us to "mix" the sets of states in the definition of stochastic bisimulation.

Definition 5.8 *Let G and G' be dynamic expressions. An* equivalence *relation $\mathcal{R} \subseteq (DR(G) \cup DR(G'))^2$ is a* $\star$-stochastic bisimulation *between G and G', $\star \in$*{interleaving, step}*, denoted by $\mathcal{R} : G \underline{\leftrightarrow}_{\star s} G'$, $\star \in \{i,s\}$, if:*

1. *$([G]_\approx, [G']_\approx) \in \mathcal{R}$.*
2. *$(s_1,s_2) \in \mathcal{R} \ \Rightarrow\ \forall \mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$*
   - *$\forall x \in \mathcal{L}$ and $\hookrightarrow = \rightharpoonup$, if $\star = i$;*
   - *$\forall x \in I\!N_{fin}^{\mathcal{L}} \setminus \{\emptyset\}$ and $\hookrightarrow = \twoheadrightarrow$, if $\star = s$;*

$$s_1 \stackrel{x}{\hookrightarrow}_{\mathcal{P}} \mathcal{H} \ \Leftrightarrow\ s_2 \stackrel{x}{\hookrightarrow}_{\mathcal{P}} \mathcal{H}.$$

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

Let $\mathcal{R}_{\star s}(G,G') = \bigcup\{\mathcal{R} \mid \mathcal{R} : G \underline{\leftrightarrow}_{\star s} G'\}$, $\star \in \{i,s\}$, be the *union of all $\star$-stochastic bisimulations* between G and G', $\star \in${*interleaving, step*}. The following proposition demonstrates that the relation $\mathcal{R}_{\star s}(G,G')$ is also an *equivalence* and $\mathcal{R}_{\star s}(G,G') : G \underline{\leftrightarrow}_{\star s} G'$, $\star \in \{i,s\}$.

Proposition 5.2 *Let G and G' be dynamic expressions and $G \underline{\leftrightarrow}_{\star s} G'$, $\star \in \{i,s\}$. Then $\mathcal{R}_{\star s}(G,G')$ is the largest $\star$-stochastic bisimulation between G and G', $\star \in$*{interleaving, step}.

Proof. See Appendix A.2. □

In [5], an algorithm for strong probabilistic bisimulation on labeled probabilistic transition systems (a reformulation of probabilistic automata) was proposed with time complexity $O(n^2m)$, where n is the number of states and m is the number of transitions. In [19], a decision algorithm for strong probabilistic bisimulation on generative labeled probabilistic transition systems was constructed with time complexity $O(m \log n)$ and space complexity $O(m+n)$. In [44], a polynomial algorithm for strong probabilistic bisimulation on probabilistic automata was presented. The mentioned algorithms for interleaving probabilistic bisimulation equivalence can be adapted for $\underline{\leftrightarrow}_{ss}$ using the method from [72], applied to get the decidability results for step bisimulation equivalence. The method takes into account that transition systems in interleaving and step semantics differ only by availability of the additional transitions corresponding to parallel execution of activities in the latter (which is our case).

5.5 Stochastic isomorphism

Stochastic isomorphism is weaker than $=_{ts*}$. The main idea of the following definition is to collect the probabilities of all transitions between the same pair of states such that the transition labels have the same multiaction parts. We use summation, since it is the probability of the event union.

Let G be a dynamic expression and $s, \tilde{s} \in DR(G)$ such that $s \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}} \{\tilde{s}\}$. In this case, we write $s \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s}$. Thus, $\mathcal{P}$ is the overall probability to enter into the one-element set of states $\{\tilde{s}\}$ starting in s via steps with the multiaction part A. In other words, $\mathcal{P}$ is a sum of all the probabilities of steps with the multiaction part A between the states s and $\tilde{s}$.

Definition 5.9 *Let G, G' be dynamic expressions. A mapping $\beta : DR(G) \to DR(G')$ is a* stochastic isomorphism *between G and G', denoted by $\beta : G =_{sto} G'$, if*

1. *β is a bijection such that $\beta([G]_\approx) = [G']_\approx$;*
2. *$\forall s, \tilde{s} \in DR(G)\ \forall A \in I\!N_{fin}^{\mathcal{L}} \setminus \{\emptyset\}\ s \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s} \ \Leftrightarrow\ \beta(s) \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}} \beta(\tilde{s})$.*

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

5.6 Interrelations of the stochastic equivalences

Note that all the algebraic equivalences of dynamic expressions we have defined, with the exception of $\approx$, can be transferred to the net level, i.e. to the corresponding marked dts-boxes. It is possible, since, by Theorem 5.1, the transition systems without empty loops of the former and the reachability graphs without empty loops of the latter are isomorphic. In the figures with examples of dts-boxes corresponding to the expressions related by the algebraic equivalences, we shall also depict their net analogues (denoted by the same symbols).

We now intend to compare the introduced stochastic equivalences and obtain the lattice of their interrelations.

Proposition 5.3 *Let $\star \in \{i, s\}$. For dynamic expressions G and G' the following holds:*

$$G \underline{\leftrightarrow}_{\star s} G' \;\Rightarrow\; G \equiv_{\star s} G'.$$

Proof. See Appendix A.3. □

Proposition 5.4 *For dynamic expressions G and G' the following holds:*

$$G =_{ts*} G' \;\Leftrightarrow\; G =_{ts} G'.$$

Proof. ($\Leftarrow$) It is enough to note that the abstraction from empty loops is based on transition probabilities which are the same for isomorphic transition systems.

($\Rightarrow$) Note that $TS(G)$ and $TS^*(G)$ (as well as $TS(G')$ and $TS^*(G')$) differ by presence of empty loops and by values of transition probabilities only. The sets of states, the labeling area, the non-empty multisets of activities which label the transitions and the initial states coincide. We have isomorphism of $TS^*(G)$ and $TS^*(G')$. For a state s of $TS^*(G)$, let s' be the state of $TS^*(G')$ such that these two states are related by the isomorphism of $TS^*(G)$ and $TS^*(G')$. Then $Exec(s) = \{\Gamma \mid \exists \tilde{s}\; s \stackrel{\Gamma}{\twoheadrightarrow} \tilde{s}\} \cup \{\emptyset\} = \{\Gamma \mid \exists \tilde{s}'\; s' \stackrel{\Gamma}{\twoheadrightarrow} \tilde{s}'\} \cup \{\emptyset\} = Exec(s')$. Note that in the previous equality we can always find the pairs of states s and s' related by the isomorphism of $TS^*(G)$ and $TS^*(G')$. Further, the definition of $PT(\Gamma, s)$ depends on $Exec(s)$ only rather than on concrete s. Thus, for each state s of $TS(G)$ the probabilities of outgoing transitions will be the same as for the corresponding state s' of $TS(G')$. Hence, $TS(G)$ and $TS(G')$ are isomorphic. □

Note that, though isomorphism of transition systems with and without empty loops appears to be the same relation, the equivalences defined on these two types of transition systems could be different. This is the case when the relations abstract from concrete activities which can occur (more exactly, from their probability parts) and take into account the overall probabilities to execute multiactions only. It is clear that the equivalences defined through transition systems with empty loops imply the relations based on those without empty loops, but the reverse implication is not valid.

For instance, we have defined stochastic isomorphism with the use of transition systems without empty loops. We can define the corresponding relation based on transition systems with empty loops as well. Then the latter equivalence will be strictly stronger than the former. As mentioned above, we have decided to abstract from empty loops because of the difficulties with infinite internal behavior. We now can give another reason for this decision: the equivalences based on transition systems with empty loops are somewhat unusual. The following example explains why.

Example 5.5 *Let $E = (\{a\}, \frac{1}{2})$, $E' = (\{a\}, \frac{1}{2})_1[](\{a\}, \frac{1}{2})_2$ and $E'' = (\{a\}, \frac{1}{3})_1[](\{a\}, \frac{1}{3})_2$.*

It is easy to see that (one-element) multisets of activities which label the transitions of $TS^*(\overline{E})$, $TS^*(\overline{E'})$, $TS^*(\overline{E''})$, and non-empty transitions (i.e. those which are not empty loops) of $TS(\overline{E})$, $TS(\overline{E'})$, $TS(\overline{E''})$, have the same multiaction part $\{\{a\}\}$.

Then $\overline{E} =_{sto} \overline{E'} =_{sto} \overline{E''}$, since the probability of the only one non-empty transition in $TS^*(\overline{E})$ is 1, the probability of both non-empty transitions in $TS^*(\overline{E'})$ and $TS^*(\overline{E''})$ is $\frac{1}{2}$, and $1 = \frac{1}{2} + \frac{1}{2}$.

On the other hand, $\overline{E}$ is not equivalent to $\overline{E'}$ with respect to the stronger version of stochastic isomorphism, since the probability of the only one non-empty transition in $TS(\overline{E})$ is $\frac{1}{2}$, whereas the probability of both non-empty transitions in $TS(\overline{E'})$ is $\frac{1}{3}$, and $\frac{1}{2} \neq \frac{2}{3} = \frac{1}{3} + \frac{1}{3}$.

In addition, $\overline{E'}$ is not equivalent to $\overline{E''}$ with respect to the stronger version of stochastic isomorphism, since the probability of both non-empty transitions in $TS(\overline{E'})$ is $\frac{1}{3}$, whereas the probability of both non-empty transitions in $TS(\overline{E''})$ is $\frac{1}{4}$, and $\frac{1}{3} + \frac{1}{3} = \frac{2}{3} \neq \frac{1}{2} = \frac{1}{4} + \frac{1}{4}$.

On the other hand, $\overline{E}$ is equivalent to $\overline{E''}$ with respect to the stronger version of stochastic isomorphism, since the probability of the only one non-empty transition in $TS(\overline{E})$ is $\frac{1}{2}$, the probability of both non-empty transitions in $TS(\overline{E''})$ is $\frac{1}{4}$, and $\frac{1}{2} = \frac{1}{4} + \frac{1}{4}$.

The transition systems with and without empty loops of $\overline{E}$, $\overline{E'}$ and $\overline{E''}$ are presented in Figure 13.

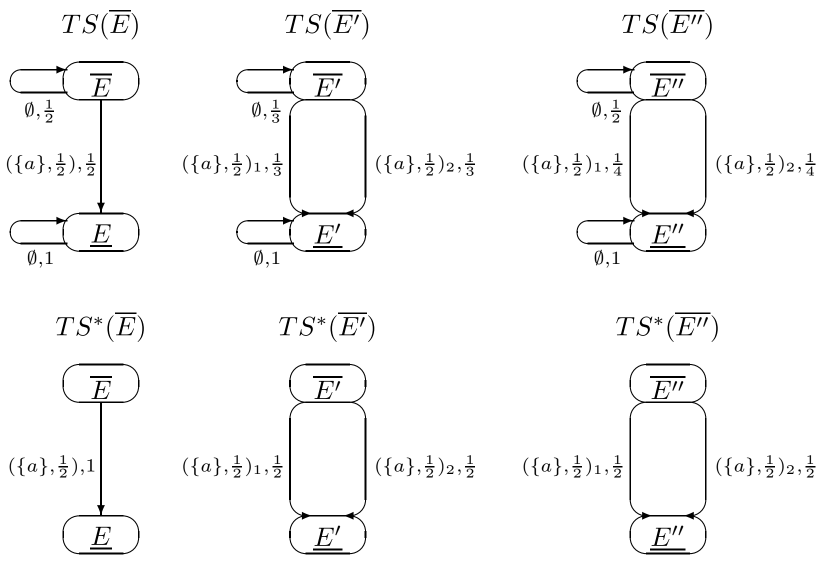

Figure 13: Properties of the stochastic isomorphism based on transition systems with empty loops

In the continuous time setting of *sPBC* there are no problems with equivalences like in the example above, but only interleaving relations can be introduced. On the other hand, the concurrency information from expressions has to be preserved in their transition systems to define correctly the congruence relation [103, 104, 107].

In the following, the symbol '_' will denote "nothing", and the equivalences subscribed by it are considered as those without any subscription such as '*is*', '*ss*', '*sto*' or '*ts*'.

Theorem 5.2 *Let* $\leftrightarrow, \leftrightarrow\!\!\!\!\leftrightarrow \in \{\equiv, \underline{\leftrightarrow}, =, \approx\}$ *and* $\star, \star\star \in \{\_, is, ss, sto, ts\}$*. For dynamic expressions* G *and* G'

$$G \leftrightarrow_\star G' \ \Rightarrow\ G \leftrightarrow\!\!\!\!\leftrightarrow_{\star\star} G'$$

iff there exists a directed path from $\leftrightarrow_\star$ *to* $\leftrightarrow\!\!\!\!\leftrightarrow_{\star\star}$ *in the graph in Figure 14.*

Proof. ($\Leftarrow$) Let us check the validity of implications in the graph in Figure 14.

- The implications $\leftrightarrow_{ss} \rightarrow \leftrightarrow_{is},\ \leftrightarrow \in \{\equiv, \underline{\leftrightarrow}\}$ are valid, since single activities are one-element multisets.
- The implications $\underline{\leftrightarrow}_{\star s} \rightarrow \equiv_{\star s},\ \star \in \{i, s\}$, are valid by Proposition 5.3.
- The implication $=_{sto} \rightarrow \underline{\leftrightarrow}_{ss}$ is proved as follows. Let $\beta : G =_{sto} G'$. Then it is easy to see that $\mathcal{S} : G \underline{\leftrightarrow}_{ss} G'$, where $\mathcal{S} = \{(s, \beta(s)) \mid s \in DR(G)\}$.
- The implication $=_{ts} \rightarrow =_{sto}$ is valid, since stochastic isomorphism is that of transition systems without empty loops up to merging of transitions with labels having identical multiaction parts.
- The implication $\approx \rightarrow =_{ts}$ is valid, since the transition system of a dynamic formula is defined based on its structural equivalence class.

($\Rightarrow$) The absence of additional nontrivial arrows (not resulting from the combination of the existing ones by transitivity) in the graph in Figure 14 is proved by the following examples.

(a) Let $E = (\{a\}, \frac{1}{2}) \| (\{b\}, \frac{1}{2})$ and $E' = ((\{a\}, \frac{1}{2}); (\{b\}, \frac{1}{2}))[]((\{b\}, \frac{1}{2}); (\{a\}, \frac{1}{2}))$. Then $\overline{E} \underline{\leftrightarrow}_{is} \overline{E'}$, but $\overline{E} \not\equiv_{ss} \overline{E'}$, since only in $TS^*(\overline{E'})$ multiactions $\{a\}$ and $\{b\}$ cannot be executed concurrently.

(b) Let $E = (\{a\}, \frac{1}{2}); ((\{b\}, \frac{1}{2})[](\{c\}, \frac{1}{2}))$ and $E' = ((\{a\}, \frac{1}{2}); (\{b\}, \frac{1}{2}))[]((\{a\}, \frac{1}{2}); (\{c\}, \frac{1}{2}))$. Then $\overline{E} \equiv_{ss} \overline{E'}$, but $\overline{E} \underline{\not\leftrightarrow}_{is} \overline{E'}$, since only in $TS^*(\overline{E'})$ a multiaction $\{a\}$ can be executed so that no multiaction $\{b\}$ can occur afterwards.

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

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

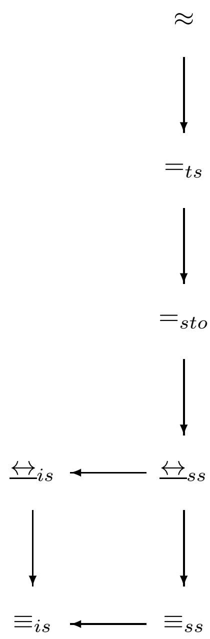

Figure 14: Interrelations of the stochastic equivalences

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

Example 5.6 *In Figure 15, the marked dts-boxes corresponding to the dynamic expressions from equivalence examples of Theorem 5.2 are presented, i.e.* $N = Box_{dts}(\overline{E})$ *and* $N' = Box_{dts}(\overline{E'})$ *for each picture (a)–(e).*

6 Reduction modulo equivalences

The equivalences which we proposed can be used to reduce transition systems and DTMCs of expressions (reachability graphs and DTMCs of dts-boxes), as well as the expressions (the dts-boxes) themselves. Reductions of graph-based models, like transition systems, reachability graphs and DTMCs, result in those with less states (the graph nodes). A reduction of expressions should result in the shorter ones with simpler structure, i.e. to those having less operators and activities. 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 properties. Thus, the reduction allows one to simplify the behavioural and performance analysis of systems.

The following example demonstrates how the stochastic equivalences can be used to simplify process expressions. Accordingly, the net analogues of the relations can be used for reduction of dts-boxes.

Example 6.1

$$\text{Let } E = ((\{a\}, \tfrac{1}{2}); (\{b\}, \tfrac{1}{2}))\|((\{c\}, \tfrac{1}{2}); (\{d\}, \tfrac{1}{2})) \text{ and}$$
$$E' = ((((\{a, x\}, \tfrac{1}{2}); ((\{b, y_1\}, \tfrac{1}{2})[]((\{b, y_2\}, \tfrac{1}{2})))\|((\{a, \hat{x}\}, \tfrac{1}{2}); ((\{b, \widehat{y_2}, y_2'\}, \tfrac{1}{2})[](\{d, v_1\}, \tfrac{1}{2})))\|$$
$$((\{c, z\}, \tfrac{1}{2}); ((\{b, \widehat{y_2'}\}, \tfrac{1}{2})[](\{d, \widehat{v_1}, v_1'\}, \tfrac{1}{2})))\|((\{c, \hat{z}\}, \tfrac{1}{2}); ((\{d, \widehat{v_1'}\}, \tfrac{1}{2})[](\{d, v_2\}, \tfrac{1}{2})))\|((\{b, \widehat{y_1}\}, \tfrac{1}{4})[](\{d, \widehat{v_2}\}, \tfrac{1}{4})))$$
$$\mathsf{sy}\ x\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ y_2'\ \mathsf{sy}\ z\ \mathsf{sy}\ v_1\ \mathsf{sy}\ v_1'\ \mathsf{sy}\ v_2\ \mathsf{rs}\ x\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ y_2'\ \mathsf{rs}\ z\ \mathsf{rs}\ v_1\ \mathsf{rs}\ v_1'\ \mathsf{rs}\ v_2.$$

Then $\overline{E}\underline{\leftrightarrow}_{ss}\overline{E'}$*, but* $\overline{E} \neq_{sto} \overline{E'}$*, since* $TS^*(\overline{E'})$ *has more states than* $TS^*(\overline{E})$*. It is clear that the syntax of* E *is much simpler than that of* E'*, but both static expressions have the same semantics induced by* $\underline{\leftrightarrow}_{ss}$*. Hence,* E *is a simplification of* E' *with respect to* $\underline{\leftrightarrow}_{ss}$*.*

In Figure 16, the marked dts-boxes corresponding to the dynamic expressions above are presented, i.e. $N = Box_{dts}(\overline{E})$ *and* $N' = Box_{dts}(\overline{E'})$*. Thus,* N *is a reduction of* N' *up to the net version of* $\underline{\leftrightarrow}_{ss}$*.*

In the general case, the procedure of expressions reduction cannot be transferred smoothly from a transition systems level. The problem is that the transition system of the reduced expression in some cases can be further reduced in such a way that it will not correspond to any expression anymore. At the net level, the reduced

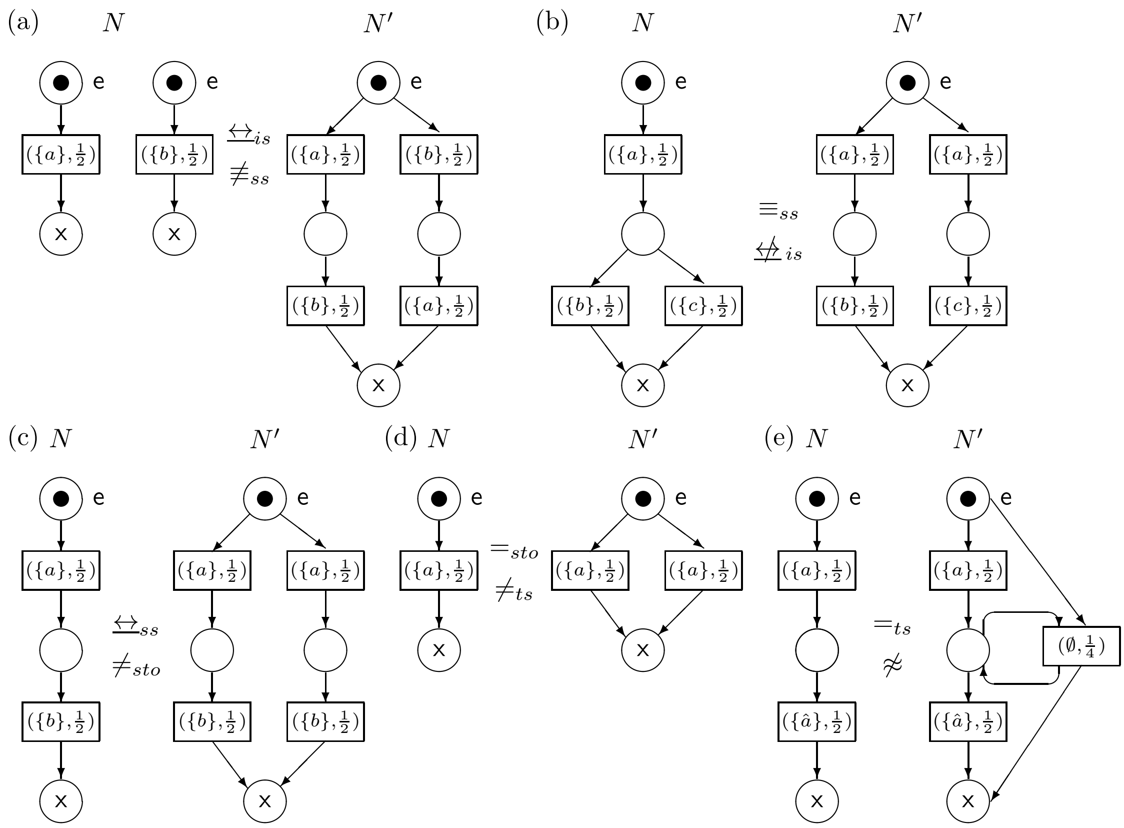

Figure 15: Dts-boxes of the dynamic expressions from equivalence examples of Theorem 5.2

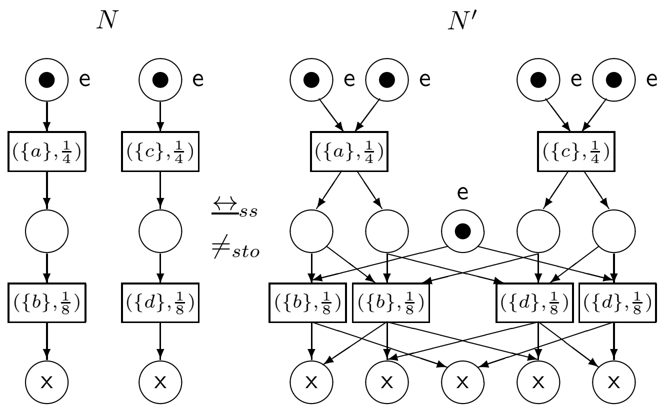

Figure 16: Reduction of a dts-box up to $\underline{\leftrightarrow}_{ss}$

transition system will be isomorphic to the reachability graph of a non-safe net which naturally cannot be a dts-box of any expression.

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 $\forall\widetilde{\mathcal{K}} \in DR(G)/_{\mathcal{R}}\ \forall A \in I\!N_{fin}^{\mathcal{L}} \setminus \{\emptyset\}\ s_1 \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}} \widetilde{\mathcal{K}} \ \Leftrightarrow\ s_2 \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}} \widetilde{\mathcal{K}}$. The previous equality is valid for all $s_1, s_2 \in \mathcal{K}$, hence, we can rewrite it as $\mathcal{K} \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}} \widetilde{\mathcal{K}}$, where $\mathcal{P} = PM_A^*(\mathcal{K}, \widetilde{\mathcal{K}}) = PM_A^*(s_1, \widetilde{\mathcal{K}}) = PM_A^*(s_2, \widetilde{\mathcal{K}})$.

We write $\mathcal{K} \stackrel{A}{\twoheadrightarrow} \widetilde{\mathcal{K}}$ if $\exists\mathcal{P}\ \mathcal{K} \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}} \widetilde{\mathcal{K}}$ and $\mathcal{K} \twoheadrightarrow \widetilde{\mathcal{K}}$ if $\exists A\ \mathcal{K} \stackrel{A}{\twoheadrightarrow} \widetilde{\mathcal{K}}$.

The similar arguments allow us to use the notation $\mathcal{K} \twoheadrightarrow_{\mathcal{P}} \widetilde{\mathcal{K}}$, where $\mathcal{P} = PM^*(\mathcal{K}, \widetilde{\mathcal{K}}) = PM^*(s_1, \widetilde{\mathcal{K}}) = PM^*(s_2, \widetilde{\mathcal{K}})$.

The *average sojourn time in the equivalence class (w.r.t. $\mathcal{R}$) of states* $\mathcal{K}$ is

$$SJ_{\mathcal{R}}(\mathcal{K}) = \frac{1}{1 - PM(\mathcal{K}, \mathcal{K})}.$$

The *average sojourn time vector for the equivalence classes (w.r.t. $\mathcal{R}$) of states* of G, denoted by $SJ_{\mathcal{R}}$, has the elements $SJ_{\mathcal{R}}(\mathcal{K}),\ \mathcal{K} \in DR(G)/_{\mathcal{R}}$.

The *sojourn time variance in the equivalence class (w.r.t. $\mathcal{R}$) of states* $\mathcal{K}$ is

$$VAR_{\mathcal{R}}(\mathcal{K}) = \frac{PM(\mathcal{K}, \mathcal{K})}{(1 - PM(\mathcal{K}, \mathcal{K}))^2}.$$

The *sojourn time variance vector for the equivalence classes (w.r.t. $\mathcal{R}$) of states* of G, denoted by $VAR_{\mathcal{R}}$, has the elements $VAR_{\mathcal{R}}(\mathcal{K}),\ \mathcal{K} \in DR(G)/_{\mathcal{R}}$.

Let $\mathcal{R}_{ss}(G) = \bigcup\{\mathcal{R} \mid \mathcal{R} : G\underline{\leftrightarrow}_{ss}G\}$ be the *union of all step stochastic autobisimulations* on G. By Proposition 5.2, $\mathcal{R}_{ss}(G)$ 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 without empty loops and the quotient (by $\underline{\leftrightarrow}_{ss}$) underlying DTMCs without empty loops of expressions can be defined. The mentioned equivalence classes become the quotient states. Every quotient transition between two such composite states represents all steps (having the same multiaction part in case of the transition system quotient) from the first state to the second one.

Definition 6.1 *Let G be a dynamic expression. The* quotient (by $\underline{\leftrightarrow}_{ss}$) (labeled probabilistic) transition system without empty loops *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*

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

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

The quotient (by $\underline{\leftrightarrow}_{ss}$) transition systems without empty loops of static expressions can be defined as well. For $E \in RegStatExpr$, let $TS^*_{\underline{\leftrightarrow}_{ss}}(E) = TS^*_{\underline{\leftrightarrow}_{ss}}(\overline{E})$.

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

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

Definition 6.2 *Let G be a dynamic expression. The* quotient (by $\underline{\leftrightarrow}_{ss}$) underlying DTMC without empty loops *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 $\mathcal{K} \twoheadrightarrow_{\mathcal{P}} \widetilde{\mathcal{K}}$, where $\mathcal{P} = PM^*(\mathcal{K}, \widetilde{\mathcal{K}})$.*

The quotient (by $\underline{\leftrightarrow}_{ss}$) underlying DTMCs without empty loops of static expressions can be defined as well. For $E \in RegStatExpr$, let $DTMC^*_{\underline{\leftrightarrow}_{ss}}(E) = DTMC^*_{\underline{\leftrightarrow}_{ss}}(\overline{E})$.

The quotients of both transition systems without empty loops and underlying DTMCs without empty loops 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 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) [125] and those of (discrete or continuous time) Markov chains by ordinary lumping [49]. 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 [152], the algorithm from [49] can be easily adjusted to produce quotients of labeled probabilistic transition systems by the probabilistic bisimulation equivalence. In [152], the symbolic partition refinement algorithm on state space of CTMCs was proposed. The algorithm can be straightforwardly accommodated to DTMCs, interactive MCs, Markov reward models, Markov decision processes, 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 [53], a polynomial-time algorithm for minimizing behaviour of probabilistic automata by probabilistic bisimulation equivalence was outlined that results in the canonical quotient structures. One could adapt the above algorithms for our framework of transition systems and DTMCs.

The comprehensive quotient examples will be presented in Section 10.

7 Logical characterization

In this section, a logical characterization of stochastic bisimulation equivalences is accomplished via formulas of probabilistic modal logics. The results obtained could be interpreted as an operational characterization of the corresponding logical equivalences. Dynamic expressions are considered as logically equivalent if they satisfy the same formulas.

7.1 Logic $iPML$

The probabilistic modal logic PML has been introduced in [87] on probabilistic transition systems without invisible actions for logical interpretation of the interleaving probabilistic bisimulation equivalence. On the basis of PML, we propose a new interleaving modal logic $iPML$ used for characterization of the interleaving stochastic bisimulation equivalence.

Definition 7.1 *Let $\top$ denote the truth and $\alpha \in \mathcal{L}$, $\mathcal{P} \in (0;1]$. A* formula *of $iPML$ is defined as follows:*

$$\Phi ::= \ \top \mid \neg\Phi \mid \Phi \wedge \Phi \mid \nabla_\alpha \mid \langle\alpha\rangle_\mathcal{P}\Phi.$$

We define $\langle\alpha\rangle\Phi = \exists\mathcal{P}\ \langle\alpha\rangle_\mathcal{P}\Phi$.

iPML denotes the set of *all formulas of the logic $iPML$.*

Definition 7.2 *Let G be a dynamic expression and $s \in DR(G)$. The* satisfaction relation $\models_G \subseteq DR(G) \times$ **iPML** *is defined as follows:*

1. *$s \models_G \top$ — always;*
2. *$s \models_G \neg\Phi$, if $s \not\models_G \Phi$;*
3. *$s \models_G \Phi \wedge \Psi$, if $s \models_G \Phi$ and $s \models_G \Psi$;*
4. *$s \models_G \nabla_\alpha$, if not $s \stackrel{\alpha}{\twoheadrightarrow} DR(G)$;*
5. *$s \models_G \langle\alpha\rangle_\mathcal{P}\Phi$, if $\exists\mathcal{H} \subseteq DR(G)\ s \stackrel{\alpha}{\twoheadrightarrow}_\mathcal{Q} \mathcal{H}$, $\mathcal{Q} \geq \mathcal{P}$ and $\forall\tilde{s} \in \mathcal{H}\ \tilde{s} \models_G \Phi$.*

Note that $\langle\alpha\rangle_\mathcal{Q}\Phi$ implies $\langle\alpha\rangle_\mathcal{P}\Phi$, if $\mathcal{Q} \geq \mathcal{P}$.

Definition 7.3 *We write $G \models_G \Phi$, if $[G]_\approx \models_G \Phi$. Two dynamic expressions G and G' are* logically equivalent *in $iPML$, denoted by $G =_{iPML} G'$, if $\forall\Phi \in$ **iPML** $G \models_G \Phi \ \Leftrightarrow\ G' \models_{G'} \Phi$.*

Let G be a dynamic expression and $s \in DR(G)$, $\alpha \in \mathcal{L}$. The set of states reached from s by execution of multiaction α, the *image set*, is defined as $Image(s,\alpha) = \{\tilde{s} \mid \exists\{(\alpha,\rho)\} \in Exec(s)\ s \stackrel{(\alpha,\rho)}{\twoheadrightarrow} \tilde{s}\}$. A dynamic expression G is an *image-finite* one, if $\forall s \in DR(G)\ \forall\alpha \in \mathcal{L}\ |Image(s,\alpha)| < \infty$.

Theorem 7.1 *For image-finite dynamic expressions G and G'*

$$G \underline{\leftrightarrow}_{is} G' \ \Leftrightarrow\ G =_{iPML} G'.$$

Proof. As the subsequent Theorem 7.2, but with state changes due to execution of single multiactions and the interleaving transition relation. □

Hence, in the interleaving semantics, we obtained a logical characterization of the stochastic bisimulation relation or, symmetrically, an operational characterization of the probabilistic modal logic equivalence.

Example 7.1 *Let* $E = (\{a\}, \frac{1}{2}); ((\{b\}, \frac{1}{2})[](\{c\}, \frac{1}{2}))$ *and* $E' = ((\{a\}, \frac{1}{2}); (\{b\}, \frac{1}{2}))[]((\{a\}, \frac{1}{2}); (\{c\}, \frac{1}{2}))$*. Then* $\overline{E} \neq_{iPML} \overline{E'}$*, because for* $\Phi = \langle\{a\}\rangle_1\langle\{b\}\rangle_{\frac{1}{2}}\top$ *we have* $\overline{E} \models_{\overline{E}} \Phi$*, but* $\overline{E'} \not\models_{\overline{E'}} \Phi$*, since in* $TS^*(\overline{E'})$ *a multiaction* $\{a\}$ *can be executed so that no multiaction* $\{b\}$ *can occur afterwards.*

7.2 Logic $sPML$

On the basis of PML, we propose a new step modal logic $sPML$ used for characterization of the step stochastic bisimulation equivalence.

Definition 7.4 *Let* $\top$ *denote the truth and* $A \in I\!N_{fin}^{\mathcal{L}} \setminus \{\emptyset\}$*,* $\mathcal{P} \in (0; 1]$*. A* formula *of sPML is defined as follows:*

$$\Phi ::= \ \top \mid \neg\Phi \mid \Phi \wedge \Phi \mid \nabla_A \mid \langle A\rangle_{\mathcal{P}}\Phi.$$

We define $\langle A\rangle\Phi = \exists\mathcal{P}\ \langle A\rangle_{\mathcal{P}}\Phi$.

sPML denotes the set of *all formulas of the logic sPML.*

Definition 7.5 *Let G be a dynamic expression and $s \in DR(G)$. The* satisfaction relation $\models_G \subseteq DR(G) \times$ **sPML** *is defined as follows:*

1. $s \models_G \top$ *— always;*
2. $s \models_G \neg\Phi$*, if* $s \not\models_G \Phi$*;*
3. $s \models_G \Phi \wedge \Psi$*, if* $s \models_G \Phi$ *and* $s \models_G \Psi$*;*
4. $s \models_G \nabla_A$*, if not* $s \stackrel{A}{\twoheadrightarrow} DR(G)$*;*
5. $s \models_G \langle A\rangle_{\mathcal{P}}\Phi$*, if* $\exists\mathcal{H} \subseteq DR(G)\ s \stackrel{A}{\twoheadrightarrow}_{\mathcal{Q}} \mathcal{H}$*,* $\mathcal{Q} \geq \mathcal{P}$ *and* $\forall\tilde{s} \in \mathcal{H}\ \tilde{s} \models_G \Phi$*.*

Note that $\langle A\rangle_{\mathcal{Q}}\Phi$ implies $\langle A\rangle_{\mathcal{P}}\Phi$, if $\mathcal{Q} \geq \mathcal{P}$.

Definition 7.6 *We write* $G \models_G \Phi$*, if* $[G]_\approx \models_G \Phi$*. Two dynamic expressions G and G' are* logically equivalent *in sPML, denoted by* $G =_{sPML} G'$*, if* $\forall\Phi \in$ **sPML** $G \models_G \Phi \ \Leftrightarrow\ G' \models_{G'} \Phi$*.*

Let G be a dynamic expression and $s \in DR(G)$, $A \in I\!N_{fin}^{\mathcal{L}} \setminus \{\emptyset\}$. The set of states reached from s by execution of a multiset of multiactions A, the *image set*, is defined as $Image(s, A) = \{\tilde{s} \mid \exists\Gamma \in Exec(s)\ \mathcal{L}(\Gamma) = A,\ s \stackrel{\Gamma}{\twoheadrightarrow} \tilde{s}\}$. A dynamic expression G is an *image-finite* one, if $\forall s \in DR(G)\ \forall A \in I\!N_{fin}^{Act}\ |Image(s, A)| < \infty$.

Theorem 7.2 *For image-finite dynamic expressions G and G'*

$$G \underline{\leftrightarrow}_{ss} G' \ \Leftrightarrow\ G =_{sPML} G'.$$

Proof. ($\Leftarrow$) To simplify the presentation, we propose the *indicator function* Ξ that recovers a dynamic expression by a state belonging to its derivation set. For a dynamic expression G and $s \in DR(G)$ we define $\Xi(s) = G$.

Let us define the equivalence relation $\mathcal{R} = \{(s_1, s_2) \in (DR(G) \cup DR(G'))^2 \mid \forall\Phi \in$ **sPML** $s_1 \models_{\Xi(s_1)} \Phi \ \Leftrightarrow\ s_2 \models_{\Xi(s_2)} \Phi\}$. We have $([G]_\approx, [G']_\approx) \in \mathcal{R}$. Let us prove that $\mathcal{R}$ is a step stochastic bisimulation.

Assume that $[G]_\approx \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}} \mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$. Let $[G']_\approx \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}'_1} s'_1, \ldots, [G']_\approx \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}'_i} s'_i, [G']_\approx \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}'_{i+1}} s'_{i+1}, \ldots, [G']_\approx \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}'_n} s'_n$ be the changes of the state $[G']_\approx$ as a result of executing the multiset of multiactions A. Since the dynamic expression G' is image-finite one, the number of such changes is finite. The state changes are ordered so that $s'_1, \ldots, s'_i \in \mathcal{H}$ and $s'_{i+1}, \ldots, s'_n \notin \mathcal{H}$.

Then $\exists\Phi_{i+1}, \ldots, \Phi_n \in$ **sPML** such that $\forall j\ (i+1 \leq j \leq n)\ \forall s \in \mathcal{H}\ s \models_{\Xi(s)} \Phi_j$, but $s'_j \not\models_{G'} \Phi_j$. We have $[G]_\approx \models_G \langle A\rangle_{\mathcal{P}}(\wedge_{j=i+1}^n \Phi_j)$ and $[G']_\approx \models_{G'} \langle A\rangle_{\mathcal{P}'}(\wedge_{j=i+1}^n \Phi_j)$, where $\mathcal{P}' = \sum_{j=1}^i \mathcal{P}'_j$.

Assume that $\mathcal{P} > \mathcal{P}'$. Then $[G']_\approx \not\models_{G'} \langle A\rangle_{\mathcal{P}}(\wedge_{j=i+1}^n \Phi_j)$, which contradicts to $([G]_\approx, [G']_\approx) \in \mathcal{R}$. Hence, $\mathcal{P} \leq \mathcal{P}'$. Consequently, $[G']_\approx \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}'} \mathcal{H}$, where $\mathcal{P} \leq \mathcal{P}'$. By symmetry of $\mathcal{R}$, we have $\mathcal{P} \geq \mathcal{P}'$. Thus, $\mathcal{P} = \mathcal{P}'$, and $\mathcal{R}$ is a step stochastic bisimulation.

($\Rightarrow$) Let for dynamic expressions G and G' we have $G\underline{\leftrightarrow}_{ss}G'$. Then $\exists\mathcal{R} : G\underline{\leftrightarrow}_{ss}G'$ and $([G]_\approx, [G']_\approx) \in \mathcal{R}$. It is sufficient to consider only the cases ∇_A and $\langle A\rangle_\mathcal{P}\Phi$, since the remaining cases are trivial.

The case ∇_A.

Assume that $[G]_\approx \models_G \nabla_A$. Then it does not hold that $[G]_\approx \xrightarrow{A}\!\!\!\!\to DR(G)$. Hence, there exist no Γ and $\tilde{s}$ such that $[G]_\approx \xrightarrow{\Gamma}\!\!\!\!\to \tilde{s}$ and $\mathcal{L}(\Gamma) = A$. Since summing by the empty index set produces zero, the transitions from each state always lead to the states of the derivation set to which that state belongs and $([G]_\approx, [G']_\approx) \in \mathcal{R}$, we get $0 = \sum_{\{\Gamma\mid\exists\tilde{s}\in DR(G)\ [G]_\approx\xrightarrow{\Gamma}\tilde{s},\ \mathcal{L}(\Gamma)=A\}} PT^*(\Gamma, [G]_\approx) = PM_A^*([G]_\approx, DR(G)) = PM_A^*([G]_\approx, DR(G) \cup DR(G')) = \sum_{\mathcal{H}\in(DR(G)\cup DR(G'))/_\mathcal{R}} PM_A^*([G]_\approx, \mathcal{H}) = \sum_{\mathcal{H}\in(DR(G)\cup DR(G'))/_\mathcal{R}} PM_A^*([G']_\approx, \mathcal{H}) = PM_A^*([G']_\approx, DR(G) \cup DR(G')) = PM_A^*([G']_\approx, DR(G')) = \sum_{\{\Gamma'\mid\exists\tilde{s}'\in DR(G')\ [G']_\approx\xrightarrow{\Gamma'}\tilde{s}',\ \mathcal{L}(\Gamma')=A\}} PT^*(\Gamma', [G']_\approx)$. Hence, there exist no Γ' and $\tilde{s}'$ such that $[G']_\approx \xrightarrow{\Gamma'}\!\!\!\!\to \tilde{s}'$ and $\mathcal{L}(\Gamma') = A$. Thus, it does not hold that $[G']_\approx \xrightarrow{A}\!\!\!\!\to DR(G')$ and we have $[G']_\approx \models_{G'} \nabla_A$.

The case $\langle A\rangle_\mathcal{P}\Phi$.

Assume that $[G]_\approx \models_G \langle A\rangle_\mathcal{P}\Phi$. Then $\exists\mathcal{H} \subseteq DR(G)$ such that $[G]_\approx \xrightarrow{A}\!\!\!\!\to_\mathcal{Q} \mathcal{H}$, $\mathcal{Q} \ge \mathcal{P}$ and $\forall s \in \mathcal{H}\ s \models_{\Xi(s)} \Phi$. Let us define $\widetilde{\mathcal{H}} = \bigcup\{\overline{\mathcal{H}} \in (DR(G) \cup DR(G'))/_\mathcal{R} \mid \overline{\mathcal{H}} \cap \mathcal{H} \ne \emptyset\}$. Then $\forall\tilde{s} \in \widetilde{\mathcal{H}}\ \exists s \in \mathcal{H}\ (s, \tilde{s}) \in \mathcal{R}$. Since $\forall s \in \mathcal{H}\ s \models_{\Xi(s)} \Phi$, we have $\forall\tilde{s} \in \widetilde{\mathcal{H}}\ \tilde{s} \models_{\Xi(\tilde{s})} \Phi$ by the induction hypothesis.

Since $\mathcal{H} \subseteq \widetilde{\mathcal{H}}$, we get $[G]_\approx \xrightarrow{A}\!\!\!\!\to_{\widetilde{\mathcal{Q}}} \widetilde{\mathcal{H}}$, $\widetilde{\mathcal{Q}} \ge \mathcal{Q}$. Since $\widetilde{\mathcal{H}}$ is the union of the equivalence classes with respect to $\mathcal{R}$, we have $([G]_\approx, [G']_\approx) \in \mathcal{R}$ implies $[G']_\approx \xrightarrow{A}\!\!\!\!\to_{\widetilde{\mathcal{Q}}} \widetilde{\mathcal{H}}$. Since $\widetilde{\mathcal{Q}} \ge \mathcal{Q} \ge \mathcal{P}$, we get $[G']_\approx \models_{G'} \langle A\rangle_\mathcal{P}\Phi$. Therefore, G' satisfies all the formulas which G does. By symmetry of $\mathcal{R}$, G satisfies all the formulas which G' does. Thus, the sets of satisfiable formulas for G and G' coincide. □

Hence, in the step semantics, we obtained a logical characterization of the stochastic bisimulation relation or, symmetrically, an operational characterization of the probabilistic modal logic equivalence.

Example 7.2 *Let* $E = (\{a\}, \frac{1}{2})\|(\{b\}, \frac{1}{2})$ *and* $E' = ((\{a\}, \frac{1}{2}); (\{b\}, \frac{1}{2}))[]((\{b\}, \frac{1}{2}); (\{a\}, \frac{1}{2}))$. *Then* $\overline{E}\underline{\leftrightarrow}_{is}\overline{E'}$ *but* $\overline{E} \ne_{sPML} \overline{E'}$, *because for* $\Phi = \langle\{a, b\}\rangle_{\frac{1}{3}}\top$ *we have* $\overline{E} \models_{\overline{E}} \Phi$, *but* $\overline{E'} \not\models_{\overline{E'}} \Phi$, *since only in* $TS^*(\overline{E'})$ *multiactions* $\{a\}$ *and* $\{b\}$ *cannot be executed concurrently.*

8 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. Let us define the expression $\mathsf{Stop} = (\{g\}, \frac{1}{2})$ rs g specifying the special process analogous to the one used in the examples of [103, 104, 107]. The latter is a continuous time stochastic analogue of the stop process proposed in [15]. Stop is a discrete time stochastic analogue of the stop, it is only able to perform empty loops with probability 1 and never terminates. Note that in the specification of Stop one could use an arbitrary action from $\mathcal{A}$ and any probability belonging to the interval $(0; 1)$. In particular, if the body of iteration contains the Stop expression, then the iteration will be "broken". On the other hand, the iteration body can be left after a finite number of its repeated executions and then the iteration termination is started. To avoid executing any activities after the iteration body, we take Stop as the termination argument of iteration.

Like in the framework of DTMCs, in DTSPNs the most common systems for performance analysis are *ergodic* (irreducible, positive recurrent and aperiodic) ones. For ergodic DTSPNs, the steady-state marking probabilities exist and can be determined. In [99], 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).

Consider dts-box of a dynamic expression $G = \overline{[E * F * \mathsf{Stop}]}$ specifying a process for which we assume that it has no deadlocks while (repetitive) running the body F of the iteration operator. Then the three ergodicity conditions are satisfied: the subnet corresponding to the looping of the iteration body F is live, safe (1-bounded) and nondeterministic (since all markings of the subnet are non-terminal, the probabilities of transitions from them are strictly less than 1). Hence, according to [97, 99], for the dts-box, its underlying DTMC, restricted to the markings of the mentioned subnet, is ergodic. The isomorphism between DTMCs of expressions and those of the corresponding dts-boxes, which is stated by Proposition 4.1, guarantees that $DTMC(G)$ is ergodic, if restricted to the states between $[[E * \overline{F} * \mathsf{Stop}]]_\approx$ and $[[E * \underline{F} * \mathsf{Stop}]]_\approx$.

However, it has been shown in [31] that even live, safe and nondeterministic DTSPNs (as well as live and safe CTSPNs and GSPNs) may be non-ergodic.

In this section, we consider only the process expressions such that their underlying DTMCs contain exactly one closed communication class of states, and this class should also 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. Remember that a communication class of states is their equivalence class w.r.t. communication relation, i.e. a maximal subset of communicating states. A communication class of states is closed if only the states belonging to it are accessible from every its state.

8.1 Theoretical background

The following methods of transient and stationary analysis are based on those from [61, 62].

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

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

The transient (k-step, $k \in \mathbb{N}$) PMF $\psi^*[k] = (\psi^*_1[k], \ldots, \psi^*_n[k])$ for $DTMC^*(G)$ is calculated as

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

where $\psi^*[0] = (\psi^*_1[0], \ldots, \psi^*_n[0])$ is the initial PMF defined as

$$\psi^*_i[0] = \begin{cases} 1, & s_i = [G]_\approx; \\ 0, & \text{otherwise.} \end{cases}$$

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

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

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

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

Note that the vector ψ^* exists and is unique, if $DTMC^*(G)$ is ergodic. Then $DTMC^*(G)$ has a single steady state, and we have $\psi^* = \lim_{k\to\infty} \psi^*[k]$. We shall consider only Markov chains with at most one steady state.

For $s \in DR(G)$ with $s = s_i$ $(1 \leq i \leq n)$ we define $\psi^*[k](s) = \psi^*_i[k]$ $(k \in \mathbb{N})$ and $\psi^*(s) = \psi^*_i$.

Let G be a dynamic expression and $s, \tilde{s} \in DR(G)$, $S, \widetilde{S} \subseteq DR(G)$. The following standard *performance indices (measures)* can be calculated based on the steady-state PMF ψ^* for $DTMC^*(G)$ [97, 99].

- The *average recurrence (return) time in the state s* (i.e. the number of discrete time units or steps required for this) is $\frac{1}{\psi^*(s)}$.
- The *fraction of residence time in the state s* is $\psi^*(s)$.
- The *fraction of residence time in the set of states S* or the *probability of the event determined by a condition that is true for all states from S* is $\sum_{s\in S} \psi^*(s)$.
- The *relative fraction of residence time in the set of states S with respect to that in* $\widetilde{S}$ is $\frac{\sum_{s\in S} \psi^*(s)}{\sum_{\tilde{s}\in \widetilde{S}} \psi^*(\tilde{s})}$.
- The *steady-state probability to perform a step with a multiset of activities* Δ is $\sum_{s\in DR(G)} \psi^*(s) \sum_{\{\Gamma \mid \Delta\subseteq\Gamma\}} PT^*(\Gamma, s)$.
- The *probability of the event determined by a reward function r on the states* is $\sum_{s\in DR(G)} \psi^*(s)r(s)$, where $\forall s \in DR(G)\ 0 \leq r(s) \leq 1$.

We have intentionally decided to evaluate performance of the modeled systems with the use of the underlying DTMCs without empty loops of the corresponding expressions. This allows us to identify the expressions up to the equivalences defined on the basis of their transition systems without empty loops. Nevertheless, from the theoretical viewpoint, it is interesting to determine a relationship between steady-state PMFs for the underlying

DTMCs with and without empty loops. The following theorem proposes the equation that relates the mentioned steady-state PMFs.

First, we introduce some helpful notation. For a vector $v = (v_1, \ldots, v_n)$, let $Diag(v)$ be a diagonal matrix of order n with the elements $Diag_{ij}(v)$ $(1 \leq i, j \leq n)$ defined as

$$Diag_{ij}(v) = \begin{cases} v_i, & i = j; \\ 0, & \text{otherwise.} \end{cases}$$

Theorem 8.1 *Let G be a dynamic expression and EL be its empty loops abstraction vector. Then the steady-state PMFs ψ for $DTMC(G)$ and ψ^* for $DTMC^*(G)$ are related as follows: $\forall s \in DR(G)$*

$$\psi(s) = \frac{\psi^*(s)EL(s)}{\sum_{\tilde{s} \in DR(G)} \psi^*(\tilde{s})EL(\tilde{s})}.$$

Proof. Note that the TPM $\mathbf{P}$ and the steady-state PMF ψ for $DTMC(G)$ are defined like the corresponding notions $\mathbf{P}^*$ and ψ^* for $DTMC^*(G)$.

Let $PT(\emptyset)$ be a vector with the elements $PT(\emptyset, s),\ s \in DR(G)$. By definition of $PM^*(s, \tilde{s})$, we have $\mathbf{P}^* = Diag(EL)(\mathbf{P} - Diag(PT(\emptyset)))$. Further,

$$\psi^*(\mathbf{P}^* - \mathbf{I}) = \mathbf{0} \text{ and } \psi^*\mathbf{P}^* = \psi^*.$$

After replacement of $\mathbf{P}^*$ by $Diag(EL)(\mathbf{P} - Diag(PT(\emptyset)))$ we obtain

$$\psi^*Diag(EL)(\mathbf{P} - Diag(PT(\emptyset))) = \psi^* \text{ and } \psi^*Diag(EL)\mathbf{P} = \psi^*(Diag(EL)Diag(PT(\emptyset)) + \mathbf{I}).$$

Note that $\forall s \in DR(G)\ EL(s)PT(\emptyset, s) + 1 = \frac{PT(\emptyset,s)}{1-PT(\emptyset,s)} + 1 = \frac{1}{1-PT(\emptyset,s)} = EL(s)$, hence, $Diag(EL)Diag(PT(\emptyset)) + \mathbf{I} = Diag(EL)$. Thus,

$$\psi^*Diag(EL)\mathbf{P} = \psi^*Diag(EL).$$

Then, for $v = \psi^*Diag(EL)$, we have

$$v\mathbf{P} = v \text{ and } v(\mathbf{P} - \mathbf{I}) = \mathbf{0}.$$

In order to calculate ψ on the basis of v, we must normalize it by dividing its elements by their sum, since we should have $\psi\mathbf{1}^T = 1$ as a result:

$$\psi = \frac{1}{v\mathbf{1}^T}v = \frac{1}{\psi^*Diag(EL)\mathbf{1}^T}\psi^*Diag(EL).$$

Thus, the elements of ψ are calculated as follows: $\forall s \in DR(G)$

$$\psi(s) = \frac{\psi^*(s)EL(s)}{\sum_{\tilde{s} \in DR(G)} \psi^*(\tilde{s})EL(\tilde{s})}.$$

It is easy to check that ψ is a solution of the equation system

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

hence, it is indeed the steady-state PMF for $DTMC(G)$. □

8.2 Steady state and equivalences

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

Proposition 8.1 *Let G, G' be dynamic expressions with $\mathcal{R} : G\underline{\leftrightarrow}_{ss}G'$ and ψ^* be the steady-state PMF for $DTMC^*(G)$, ψ'^* be the steady-state PMF for $DTMC^*(G')$. Then $\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$*

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

Proof. See Appendix A.4. □

Note that in the proof of Proposition 8.1 a limit construction was used to go from transient to stationary case. Thus, the result of this proposition is valid as well if we replace steady-state probabilities with transient ones in its statement.

Let G be a dynamic expression. The transient PMF $\psi^*_{\underline{\leftrightarrow}_{ss}}[k]$ $(k \in I\!N)$ and the steady-state PMF $\psi^*_{\underline{\leftrightarrow}_{ss}}$ for $DTMC^*_{\underline{\leftrightarrow}_{ss}}(G)$ are defined like the corresponding notions $\psi^*[k]$ and ψ^* for $DTMC^*(G)$.

By Proposition 8.1, we have $\forall\mathcal{K} \in DR(G)/_{\mathcal{R}_{ss}(G)}$

$$\psi^*_{\underline{\leftrightarrow}_{ss}}(\mathcal{K}) = \sum_{s\in\mathcal{K}} \psi^*(s).$$

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

The following example demonstrates that the result of Proposition 8.1 does not hold for $\underline{\leftrightarrow}_{is}$.

Example 8.1 *Let* $E = [(\{a\}, \frac{1}{2}) * ((\{b\}, \frac{1}{2}); ((\{c\}, \frac{1}{2}) \| (\{d\}, \frac{1}{2}))) * \mathsf{Stop}]$ *and*
$E' = [(\{a\}, \frac{1}{2}) * ((\{b\}, \frac{1}{2}); (((\{c\}, \frac{1}{2})_1; (\{d\}, \frac{1}{2})_1)[]((\{d\}, \frac{1}{2})_2; (\{c\}, \frac{1}{2})_2))) * \mathsf{Stop}]$. *We have* $\overline{E}\underline{\leftrightarrow}_{is}\overline{E'}$.

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

$$\begin{aligned}
s_1 &= [[\overline{(\{a\}, \tfrac{1}{2})} * ((\{b\}, \tfrac{1}{2}); ((\{c\}, \tfrac{1}{2}) \| (\{d\}, \tfrac{1}{2}))) * \mathsf{Stop}]]_{\approx},\\
s_2 &= [[(\{a\}, \tfrac{1}{2}) * (\overline{(\{b\}, \tfrac{1}{2})}; ((\{c\}, \tfrac{1}{2}) \| (\{d\}, \tfrac{1}{2}))) * \mathsf{Stop}]]_{\approx},\\
s_3 &= [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); (\overline{(\{c\}, \tfrac{1}{2})} \| \overline{(\{d\}, \tfrac{1}{2})})) * \mathsf{Stop}]]_{\approx},\\
s_4 &= [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); (\underline{(\{c\}, \tfrac{1}{2})} \| \overline{(\{d\}, \tfrac{1}{2})})) * \mathsf{Stop}]]_{\approx},\\
s_5 &= [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); (\overline{(\{c\}, \tfrac{1}{2})} \| \underline{(\{d\}, \tfrac{1}{2})})) * \mathsf{Stop}]]_{\approx}.
\end{aligned}$$

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

$$\begin{aligned}
s'_1 &= [[\overline{(\{a\}, \tfrac{1}{2})} * ((\{b\}, \tfrac{1}{2}); (((\{c\}, \tfrac{1}{2})_1; (\{d\}, \tfrac{1}{2})_1)[]((\{d\}, \tfrac{1}{2})_2; (\{c\}, \tfrac{1}{2})_2))) * \mathsf{Stop}]]_{\approx},\\
s'_2 &= [[(\{a\}, \tfrac{1}{2}) * (\overline{(\{b\}, \tfrac{1}{2})}; (((\{c\}, \tfrac{1}{2})_1; (\{d\}, \tfrac{1}{2})_1)[]((\{d\}, \tfrac{1}{2})_2; (\{c\}, \tfrac{1}{2})_2))) * \mathsf{Stop}]]_{\approx},\\
s'_3 &= [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); \overline{(((\{c\}, \tfrac{1}{2})_1; (\{d\}, \tfrac{1}{2})_1)[]((\{d\}, \tfrac{1}{2})_2; (\{c\}, \tfrac{1}{2})_2))}) * \mathsf{Stop}]]_{\approx},\\
s'_4 &= [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); (((\{c\}, \tfrac{1}{2})_1; \overline{(\{d\}, \tfrac{1}{2})_1})[]((\{d\}, \tfrac{1}{2})_2; (\{c\}, \tfrac{1}{2})_2))) * \mathsf{Stop}]]_{\approx},\\
s'_5 &= [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); (((\{c\}, \tfrac{1}{2})_1; (\{d\}, \tfrac{1}{2})_1)[]((\{d\}, \tfrac{1}{2})_2; \overline{(\{c\}, \tfrac{1}{2})_2}))) * \mathsf{Stop}]]_{\approx}.
\end{aligned}$$

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

$$\psi^* = \left(0, \frac{3}{8}, \frac{3}{8}, \frac{1}{8}, \frac{1}{8}\right),\ \psi'^* = \left(0, \frac{1}{3}, \frac{1}{3}, \frac{1}{6}, \frac{1}{6}\right).$$

Consider the equivalence class (with respect to $\mathcal{R}_{ss}(\overline{E}, \overline{E'})$) $\mathcal{H} = \{s_3, s'_3\}$. We have $\sum_{s\in\mathcal{H}\cap DR(\overline{E})} \psi^*(s) = \psi^*(s_3) = \frac{3}{8}$, whereas $\sum_{s'\in\mathcal{H}\cap DR(\overline{E'})} \psi'^*(s') = \psi'^*(s'_3) = \frac{1}{3}$. Thus, $\underline{\leftrightarrow}_{is}$ does not guarantee coincidence of steady-state probabilities to enter into an equivalence class.

In Figure 17, the marked dts-boxes corresponding to the dynamic expressions above are presented, i.e. $N = Box_{dts}(\overline{E})$ and $N' = Box_{dts}(\overline{E'})$.

The following example demonstrates that the result of Proposition 8.1 does not even hold for the intersection of $\underline{\leftrightarrow}_{is}$ and $\equiv_{ss}$.

Example 8.2 *Let* $E = [(\{a\}, \frac{1}{2}) * ((\{b\}, \frac{1}{2}); ((\{c\}, \frac{1}{2}) \| (\{d\}, \frac{1}{2}))) * \mathsf{Stop}]$ *and*
$E' = [(\{a\}, \frac{1}{2}) * ((\{b\}, \frac{1}{2}); (((\{c\}, \frac{1}{2})_1 \| (\{d\}, \frac{1}{2})_1))[](((\{c\}, \frac{1}{2})_2; (\{d\}, \frac{1}{2})_2)[]((\{d\}, \frac{1}{2})_3; (\{c\}, \frac{1}{2})_3))))) * \mathsf{Stop}]$. *We have* $\overline{E}\underline{\leftrightarrow}_{is}\overline{E'}$ *and* $\overline{E} \equiv_{ss} \overline{E'}$.

$DR(\overline{E})$ is given in the Example 8.1.

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

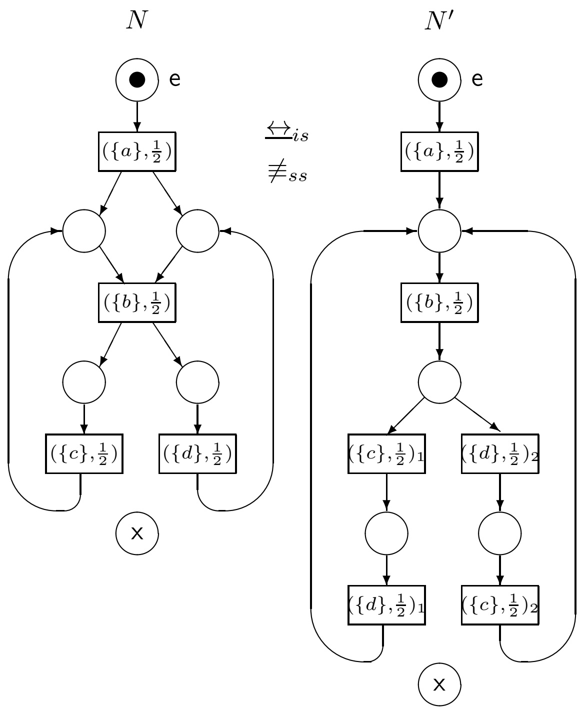

Figure 17: $\underline{\leftrightarrow}_{is}$ does not guarantee coincidence of steady-state probabilities to enter into an equivalence class

$$s'_1 = [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); (((\{c\}, \tfrac{1}{2})_1 \| (\{d\}, \tfrac{1}{2})_1))[](((\{c\}, \tfrac{1}{2})_2; (\{d\}, \tfrac{1}{2})_2)[](((\{d\}, \tfrac{1}{2})_3; (\{c\}, \tfrac{1}{2})_3))))) * \mathsf{Stop}]]_\approx,$$
$$s'_2 = [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); (((\{c\}, \tfrac{1}{2})_1 \| (\{d\}, \tfrac{1}{2})_1))[](((\{c\}, \tfrac{1}{2})_2; (\{d\}, \tfrac{1}{2})_2)[](((\{d\}, \tfrac{1}{2})_3; (\{c\}, \tfrac{1}{2})_3))))) * \mathsf{Stop}]]_\approx,$$
$$s'_3 = [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); (((\{c\}, \tfrac{1}{2})_1 \| (\{d\}, \tfrac{1}{2})_1))[](((\{c\}, \tfrac{1}{2})_2; (\{d\}, \tfrac{1}{2})_2)[](((\{d\}, \tfrac{1}{2})_3; (\{c\}, \tfrac{1}{2})_3))))) * \mathsf{Stop}]]_\approx,$$
$$s'_4 = [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); (((\{c\}, \tfrac{1}{2})_1 \| (\{d\}, \tfrac{1}{2})_1))[](((\{c\}, \tfrac{1}{2})_2; (\{d\}, \tfrac{1}{2})_2)[](((\{d\}, \tfrac{1}{2})_3; (\{c\}, \tfrac{1}{2})_3))))) * \mathsf{Stop}]]_\approx,$$
$$s'_5 = [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); (((\{c\}, \tfrac{1}{2})_1 \| (\{d\}, \tfrac{1}{2})_1))[](((\{c\}, \tfrac{1}{2})_2; (\{d\}, \tfrac{1}{2})_2)[](((\{d\}, \tfrac{1}{2})_3; (\{c\}, \tfrac{1}{2})_3))))) * \mathsf{Stop}]]_\approx,$$
$$s'_6 = [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); (((\{c\}, \tfrac{1}{2})_1 \| (\{d\}, \tfrac{1}{2})_1))[](((\{c\}, \tfrac{1}{2})_2; (\{d\}, \tfrac{1}{2})_2)[](((\{d\}, \tfrac{1}{2})_3; (\{c\}, \tfrac{1}{2})_3))))) * \mathsf{Stop}]]_\approx,$$
$$s'_7 = [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); (((\{c\}, \tfrac{1}{2})_1 \| (\{d\}, \tfrac{1}{2})_1))[](((\{c\}, \tfrac{1}{2})_2; (\{d\}, \tfrac{1}{2})_2)[](((\{d\}, \tfrac{1}{2})_3; (\{c\}, \tfrac{1}{2})_3))))) * \mathsf{Stop}]]_\approx.$$

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

$$\psi^* = \left(0, \frac{3}{8}, \frac{3}{8}, \frac{1}{8}, \frac{1}{8}\right),\ \psi'^* = \left(0, \frac{13}{38}, \frac{13}{38}, \frac{3}{38}, \frac{3}{38}, \frac{3}{38}, \frac{3}{38}\right).$$

Consider the equivalence class (with respect to $\mathcal{R}_{ss}(\overline{E}, \overline{E'})$) $\mathcal{H} = \{s_3, s'_3\}$. We have $\sum_{s \in \mathcal{H} \cap DR(\overline{E})} \psi^*(s) = \psi^*(s_3) = \frac{3}{8}$, whereas $\sum_{s' \in \mathcal{H} \cap DR(\overline{E'})} \psi'^*(s') = \psi'^*(s'_3) = \frac{13}{38}$. Thus, the intersection of $\underline{\leftrightarrow}_{is}$ and $\equiv_{ss}$ does not guarantee coincidence of steady-state probabilities to enter into an equivalence class.

In Figure 18, the marked dts-boxes corresponding to the dynamic expressions above are presented, i.e. $N = Box_{dts}(\overline{E})$ and $N' = Box_{dts}(\overline{E'})$.

By Proposition 8.1, $\underline{\leftrightarrow}_{ss}$ preserves the quantitative properties of the stationary behaviour (the level of DTMCs). 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 8.1 *A* derived step trace *of a dynamic expression G is a chain $\Sigma = A_1 \cdots A_n \in (I\!N_{fin}^{\mathcal{L}} \setminus \{\emptyset\})^*$, where $\exists s \in DR(G)\ s \stackrel{\Gamma_1}{\twoheadrightarrow} s_1 \stackrel{\Gamma_2}{\twoheadrightarrow} \cdots \stackrel{\Gamma_n}{\twoheadrightarrow} s_n,\ \mathcal{L}(\Gamma_i) = A_i\ (1 \le i \le n)$. Then the* probability to execute the derived step trace Σ in s *is*

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

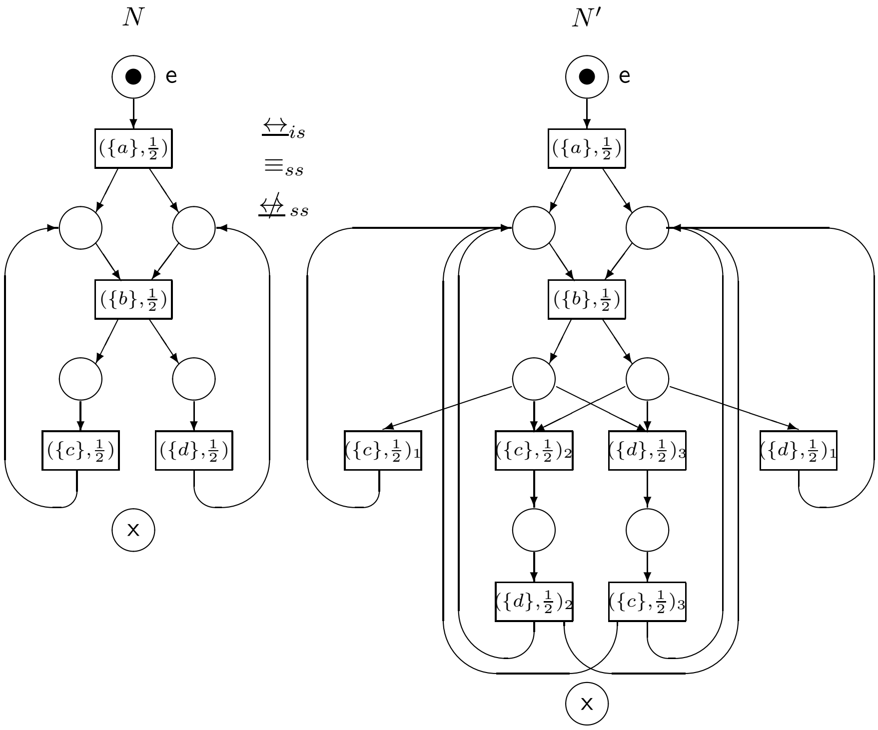

Figure 18: The intersection of $\underline{\leftrightarrow}_{is}$ and $\equiv_{ss}$ does not guarantee coincidence of steady-state probabilities to enter into an equivalence class

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

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

Proof. See Appendix A.5. □

Note that in the proof of Theorem 8.2 the result of Proposition 8.1 was used that is also valid for the transient probabilities. Thus, the result of this theorem is valid as well if we replace steady-state probabilities with transient ones in its statement.

Let G be a dynamic expression, φ^* be the steady-state PMF for $DTMC^*(G)$, $\varphi^*_{\underline{\leftrightarrow}_{ss*}}$ be the steady-state PMF for $DTMC^*_{\underline{\leftrightarrow}_{ss*}}(G)$ and Σ be a derived step trace of G. By Theorem 8.2, we have $\forall \mathcal{K} \in DR(G)/_{\mathcal{R}_{ss*}(G)}$

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

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

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 8.2 *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. See Appendix A.6. □

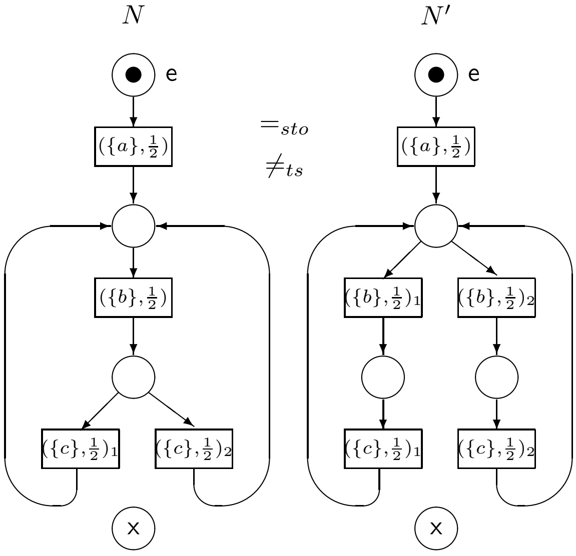

Figure 19: $\underline{\leftrightarrow}_{ss}$ preserves steady-state behaviour and sojourn time properties in the equivalence classes

Example 8.3 *Let* $E = [(\{a\},\frac{1}{2}) * ((\{b\},\frac{1}{2});((\{c\},\frac{1}{2})_1[](\{c\},\frac{1}{2})_2)) * \mathsf{Stop}]$ *and* $E' = [(\{a\},\frac{1}{2}) * (((\{b\},\frac{1}{2})_1;(\{c\},\frac{1}{2})_1)[]((\{b\},\frac{1}{2})_2;(\{c\},\frac{1}{2})_2)) * \mathsf{Stop}]$. *We have* $\overline{E} =_{sto} \overline{E'}$*, hence,* $\overline{E}\underline{\leftrightarrow}_{ss}\overline{E'}$.

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

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

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

$$\begin{aligned}
s'_1 &= [[\overline{(\{a\},\tfrac{1}{2})} * (((\{b\},\tfrac{1}{2})_1;(\{c\},\tfrac{1}{2})_1)[]((\{b\},\tfrac{1}{2})_2;(\{c\},\tfrac{1}{2})_2)) * \mathsf{Stop}]]_\approx,\\
s'_2 &= [[(\{a\},\tfrac{1}{2}) * \overline{(((\{b\},\tfrac{1}{2})_1;(\{c\},\tfrac{1}{2})_1)[]((\{b\},\tfrac{1}{2})_2;(\{c\},\tfrac{1}{2})_2))} * \mathsf{Stop}]]_\approx,\\
s'_3 &= [[(\{a\},\tfrac{1}{2}) * (((\{b\},\tfrac{1}{2})_1;\overline{(\{c\},\tfrac{1}{2})_1})[]((\{b\},\tfrac{1}{2})_2;(\{c\},\tfrac{1}{2})_2)) * \mathsf{Stop}]]_\approx,\\
s'_4 &= [[(\{a\},\tfrac{1}{2}) * (((\{b\},\tfrac{1}{2})_1;(\{c\},\tfrac{1}{2})_1)[]((\{b\},\tfrac{1}{2})_2;\overline{(\{c\},\tfrac{1}{2})_2})) * \mathsf{Stop}]]_\approx.
\end{aligned}$$

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

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

Consider the equivalence class (with respect to $\mathcal{R}_{ss}(\overline{E},\overline{E'})$) $\mathcal{H} = \{s_3, s'_3, s'_4\}$. Observe that the steady-state probabilities for $\mathcal{H}$ coincide: $\sum_{s\in\mathcal{H}\cap DR(\overline{E})}\psi^*(s) = \psi^*(s_3) = \frac{1}{2} = \frac{1}{4}+\frac{1}{4} = \psi'^*(s'_3)+\psi'^*(s'_4) = \sum_{s'\in\mathcal{H}\cap DR(\overline{E'})}\psi'^*(s')$. Let $\Sigma = \{\{c\}\}$. The steady-state probabilities to enter into the equivalence class $\mathcal{H}$ and start the step trace Σ from it coincide as well: $\psi^*(s_3)(PT^*(\{(\{c\},\frac{1}{2})_1\},s_3) + PT^*(\{(\{c\},\frac{1}{2})_2\},s_3)) = \frac{1}{2}\left(\frac{1}{2}+\frac{1}{2}\right) = \frac{1}{2} = \frac{1}{4}\cdot 1 + \frac{1}{4}\cdot 1 = \psi'^*(s'_3)PT^*(\{(\{c\},\frac{1}{2})_1\},s'_3) + \psi'^*(s'_4)PT^*(\{(\{c\},\frac{1}{2})_2\},s'_4)$.

Further, the sojourn time averages in the equivalence class $\mathcal{H}$ coincide: $SJ_{\mathcal{R}_{ss}(\overline{E},\overline{E'})\cap(DR(\overline{E}))^2}(\mathcal{H}\cap DR(G)) = SJ_{\mathcal{R}_{ss}(\overline{E},\overline{E'})\cap(DR(\overline{E}))^2}(\{s_3\}) = \frac{1}{1-PM(\{s_3\},\{s_3\})} = \frac{1}{1-PM(s_3,s_3)} = \frac{1}{1-\frac{1}{2}} = 2 = \frac{1}{1-\frac{1}{2}} = \frac{1}{1-PM(s'_3,s'_3)} = \frac{1}{1-PM(s'_4,s'_4)} = \frac{1}{1-PM(\{s'_3,s'_4\},\{s'_3,s'_4\})} = SJ_{\mathcal{R}_{ss}(\overline{E},\overline{E'})\cap(DR(\overline{E'}))^2}(\{s'_3,s'_4\}) = SJ_{\mathcal{R}_{ss}(\overline{E},\overline{E'})\cap(DR(\overline{E'}))^2}(\mathcal{H}\cap DR(G'))$.

Next, the sojourn time variances in the equivalence class $\mathcal{H}$ coincide: $VAR_{\mathcal{R}_{ss}(\overline{E},\overline{E'})\cap(DR(\overline{E}))^2}(\mathcal{H}\cap DR(G)) = VAR_{\mathcal{R}_{ss}(\overline{E},\overline{E'})\cap(DR(\overline{E}))^2}(\{s_3\}) = \frac{PM(\{s_3\},\{s_3\})}{(1-PM(\{s_3\},\{s_3\}))^2} = \frac{PM(s_3,s_3)}{(1-PM(s_3,s_3))^2} = \frac{\frac{1}{2}}{(1-\frac{1}{2})^2} = 2 = \frac{\frac{1}{2}}{(1-\frac{1}{2})^2} = \frac{PM(s'_3,s'_3)}{(1-PM(s'_3,s'_3))^2} = \frac{PM(s'_4,s'_4)}{(1-PM(s'_4,s'_4))^2} = \frac{PM(\{s'_3,s'_4\},\{s'_3,s'_4\})}{(1-PM(\{s'_3,s'_4\},\{s'_3,s'_4\}))^2} = VAR_{\mathcal{R}_{ss}(\overline{E},\overline{E'})\cap(DR(\overline{E'}))^2}(\{s'_3,s'_4\}) = VAR_{\mathcal{R}_{ss}(\overline{E},\overline{E'})\cap(DR(\overline{E'}))^2}(\mathcal{H}\cap DR(G'))$.

In Figure 19, the marked dts-boxes corresponding to the dynamic expressions above are presented, i.e. $N = Box_{dts}(\overline{E})$ and $N' = Box_{dts}(\overline{E'})$.

Figure 20: Equivalence-based simplification of performance evaluation

8.3 Preservation of performance and simplification of its analysis

Many performance indices are based on the steady-state probabilities to enter into a set of similar states or, after coming in it, to start a step trace from this set. The similarity of states is usually captured by an equivalence relation, hence, the sets are often the equivalence classes. Proposition 8.1 and Theorem 8.2 guarantee coincidence of the mentioned indices for the expressions related by $\underline{\leftrightarrow}_{ss}$. Thus, $\underline{\leftrightarrow}_{ss}$ preserves performance of stochastic systems modeled by expressions of $dtsPBC$. Moreover, Example 8.1 demonstrates that it is the weakest relation among the relations we considered that has the performance preservation property.

In addition, obviously, it is easier to evaluate performance with the use of a DTMC 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 system under investigation is specified by a static expression of $dtsPBC$.
2. The transition system without empty loops of the expression is constructed.
3. After examining this transition system for self-similarity and symmetry, a step stochastic autobisimulation equivalence for the expression is determined.
4. The quotient underlying DTMC without empty loops of the expression is constructed from the quotient transition system without empty loops.
5. The steady-state probabilities and performance indices based on this DTMC are calculated.

The limitation of the method above is its applicability only to the expressions such that their underlying DTMCs contain exactly one closed communication class of states, and this class should also be ergodic to ensure uniqueness of the stationary distribution. If a DTMC 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 DTMCs of this kind as well [84]. Note that the underlying DTMC 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 or symmetry in their behaviour.

For transition systems reduction one can also use an analogue of the approach described in [85]: first perform the fast symmetry reduction based on the method from [80], then construct a quotient of the resulting transition system by bisimulation equivalence by applying the time-optimal partition refinement algorithm from [49] to the state space of this system. As mentioned in [85], for a number of practical case studies, minimization by bisimulation results in more significant state space reduction than symmetry reduction, but the latter is much faster that the former, since symmetries are determined on a syntactical level. In [8], the effective analysis methods were proposed for partially symmetric models as well.

Figure 20 presents the main stages of the equivalence-based simplification of performance evaluation described above.

9 Preservation by algebraic operations

An important question concerning equivalence relations is whether two compound expressions always remain equivalent if they are constructed from pairwise equivalent subexpressions. The equivalence having the mentioned property of preservation by algebraic operations is called a congruence. To be a congruence is a desirable property but not an obligatory one, since many important behavioural equivalences are not congruences. As a rule, a congruence relation is too discriminate, i.e. it differentiates too many formulas. This is the reason why a weaker but more interesting equivalence notion that is not a congruence is preferred in many cases when process behaviour is to be compared.

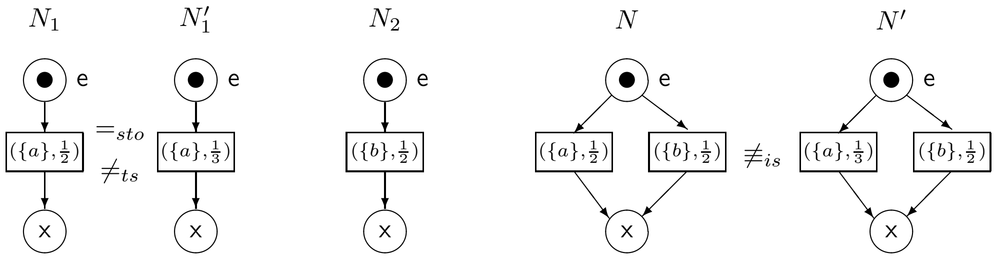

Figure 21: The equivalences between $\equiv_{is}$ and $=_{sto}$ are not congruences

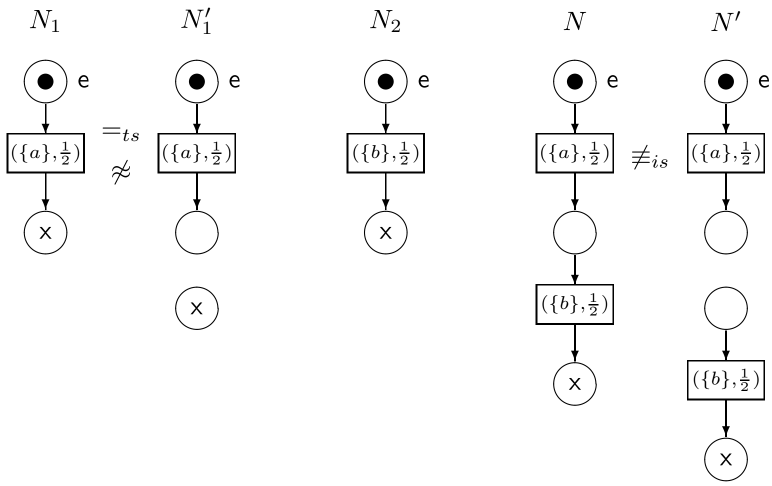

Figure 22: The equivalences between $\equiv_{is}$ and $=_{ts}$ are not congruences

Definition 9.1 *Let $\leftrightarrow$ be an equivalence of dynamic expressions. Two static expressions E and E' are* equivalent with respect to $\leftrightarrow$, *denoted by $E \leftrightarrow E'$, if $\overline{E} \leftrightarrow \overline{E'}$.*

Let us investigate which algebraic equivalences we proposed are congruences on static expressions.

The following example demonstrates that no equivalence between $\equiv_{is}$ and $=_{sto}$ is a congruence.

Example 9.1 *Let $E = (\{a\}, \frac{1}{2})$, $E' = (\{a\}, \frac{1}{3})$ and $F = (\{b\}, \frac{1}{2})$. We have $\overline{E} =_{sto} \overline{E'}$, since both $TS^*(\overline{E})$ and $TS^*(\overline{E'})$ have the transitions with the multiaction part $\{a\}$ of their labels and probability 1. On the other hand, $\overline{E[]F} \not\equiv_{is} \overline{E'[]F}$, since only in $TS^*(\overline{E'[]F})$ the probabilities of the transitions with the multiaction parts $\{a\}$ and $\{b\}$ of their labels are different ($\frac{1}{3}$ and $\frac{2}{3}$, respectively). Thus, no equivalence between $\equiv_{is}$ and $=_{sto}$ is a congruence.*

In Figure 21, the marked dts-boxes corresponding to the dynamic expressions above are presented, i.e. $N_1 = Box_{dts}(\overline{E})$, $N_1' = Box_{dts}(\overline{E'})$, $N_2 = Box_{dts}(\overline{F})$ and $N = Box_{dts}(\overline{E[]F})$, $N' = Box_{dts}(\overline{E'[]F})$.

The following proposition demonstrates that all the equivalences between $\equiv_{is}$ and $=_{ts}$ are not congruences.

Proposition 9.1 *Let $\star \in \{is, ss\}$, $\star\star \in \{sto, ts\}$. The equivalences $\equiv_\star$, $\underline{\leftrightarrow}_\star$, $=_{\star\star}$ are not preserved by algebraic operations.*

Proof. Let $E = (\{a\}, \frac{1}{2})$, $E' = (\{a\}, \frac{1}{2}); \mathsf{Stop}$ and $F = (\{b\}, \frac{1}{2})$. We have $\overline{E} =_{ts} \overline{E'}$, since both $TS(\overline{E})$ and $TS(\overline{E'})$ have the transitions with the multiaction part $\{a\}$ of their labels and probability $\frac{1}{2}$. On the other hand, $\overline{E;F} \not\equiv_{is} \overline{E';F}$, since only in $TS^*(\overline{E';F})$ no other transition can fire after the transition with the multiaction part $\{a\}$ of its label. Thus, no equivalence between $\equiv_{is}$ and $=_{ts}$ is a congruence.

In Figure 22, the marked dts-boxes corresponding to the dynamic expressions above are presented, i.e. $N_1 = Box_{dts}(\overline{E})$, $N_1' = Box_{dts}(\overline{E'})$, $N_2 = Box_{dts}(\overline{F})$ and $N = Box_{dts}(\overline{E[]F})$, $N' = Box_{dts}(\overline{E'[]F})$. □

The following proposition demonstrates that $\approx$ is a congruence.

Proposition 9.2 *The equivalence $\approx$ is preserved by algebraic operations.*

Proof. By definition of $\approx$. $\square$

We suppose that, for an analogue of $=_{ts}$ to be a congruence, we have to equip transition systems of expressions with two extra transitions skip and redo like in [103, 107]. This allows one to avoid difficulties demonstrated in the example from the proof of Proposition 9.1 with unexpected termination due to the Stop process. At the same time, such an enrichment of transition systems does not overcome the problems explained in Example 9.1 with abstraction from empty loops. Hence, the equivalences between $\equiv_{is}$ and $=_{sto}$ defined on the basis of the enriched transition systems will still be non-congruences.

To define the analogue of $=_{ts}$ mentioned above, we shall introduce a notion of sr-transition system. It has the final state and two extra transitions from the initial state to the final one and back. Note that sr-transition systems do not have the loop transitions from the final state to itself. First, in Table 7, we propose the rules for skip and redo. In this table, $E \in RegStatExpr$.

Table 7: Rules for skip and redo

$$\textbf{Sk}\ \overline{E} \stackrel{\mathsf{skip}}{\to} \underline{E} \quad \textbf{Rd}\ \underline{E} \stackrel{\mathsf{redo}}{\to} \overline{E}$$

We now can define sr-transition systems of dynamic expressions in the form $\overline{E}$, where E is a static expression. This syntactic restriction is necessary to take into account two additional rules above. We assume that skip has probability 0, hence, it will be never executed. On the other hand, redo has probability 1, hence, it will be immediately executed at the next time moment if it is enabled.

Definition 9.2 *Let E be a static expression and $TS(\overline{E}) = (S, L, \mathcal{T}, s)$. The* (labeled probabilistic) sr-transition system *of $\overline{E}$ is a quadruple $TS_{sr}(\overline{E}) = (S_{sr}, L_{sr}, \mathcal{T}_{sr}, s_{sr})$, where*

- $S_{sr} = S \cup \{[\underline{E}]_\approx\}$;
- $L_{sr} \subseteq (I\!N_{fin}^{\mathcal{SL}} \times (0;1]) \cup \{(\mathsf{skip}, 0), (\mathsf{redo}, 1)\}$;
- $\mathcal{T}_{sr} = \mathcal{T} \setminus \{([\underline{E}]_\approx, (\emptyset, 1), [\underline{E}]_\approx)\} \cup \{([\overline{E}]_\approx, (\mathsf{skip}, 0), [\underline{E}]_\approx), ([\underline{E}]_\approx, (\mathsf{redo}, 1), [\overline{E}]_\approx)\}$;
- $s_{sr} = s$.

We define a new notion of isomorphism for sr-transition systems, since we should take care of their final states.

Definition 9.3 *Let E, E' be static expressions and $TS_{sr}(\overline{E}) = (S_{sr}, L_{sr}, \mathcal{T}_{sr}, s_{sr})$, $TS_{sr}(\overline{E'}) = (S'_{sr}, L'_{sr}, \mathcal{T}'_{sr}, s'_{sr})$ be their sr-transition systems. A mapping $\beta : S_{sr} \to S'_{sr}$ is an* isomorphism *between $TS_{sr}(\overline{E})$ and $TS_{sr}(\overline{E'})$, denoted by $\beta : TS_{sr}(\overline{E}) \simeq TS_{sr}(\overline{E'})$, if*

1. *β is a bijection such that $\beta(s_{sr}) = s'_{sr}$ and $\beta([\underline{E}]_\approx) = [\underline{E'}]_\approx$;*
2. *$\forall s, \tilde{s} \in S_{sr}\ \forall \Gamma\ s \stackrel{\Gamma}{\to}_{\mathcal{P}} \tilde{s} \ \Leftrightarrow\ \beta(s) \stackrel{\Gamma}{\to}_{\mathcal{P}} \beta(\tilde{s})$.*

Two sr-transition systems $TS_{sr}(\overline{E})$ and $TS_{sr}(\overline{E'})$ are isomorphic*, denoted by $TS_{sr}(\overline{E}) \simeq TS_{sr}(\overline{E'})$, if $\exists \beta : TS_{sr}(\overline{E}) \simeq TS_{sr}(\overline{E'})$.*

sr-transition systems of static expressions can be defined as well. For $E \in RegStatExpr$, let $TS_{sr}(E) = TS_{sr}(\overline{E})$.

Example 9.2 *Let $E = (\{a\}, \frac{1}{2})$. In Figure 23, the transition systems $TS_{sr}(\overline{E})$ and $TS_{sr}(\overline{E; \mathsf{Stop}})$ are presented. In the latter sr-transition system (unlike the former one) the final state can be reached by executing the transition* $(\mathsf{skip}, 0)$ *only from the initial state.*

Definition 9.4 *Two dynamic expressions $\overline{E}$ and $\overline{E'}$ are* equivalent with respect to sr-transition systems*, denoted by $\overline{E} =_{tssr} \overline{E'}$, if $TS_{sr}(\overline{E}) \simeq TS_{sr}(\overline{E'})$.*

Note that sr-transition systems without empty loops can be defined, as well as the equivalence $=_{tssr*}$ based on them. At the same time, the coincidence of $=_{tssr}$ and $=_{tssr*}$ can be proved similar to that of $=_{ts}$ and $=_{ts*}$.

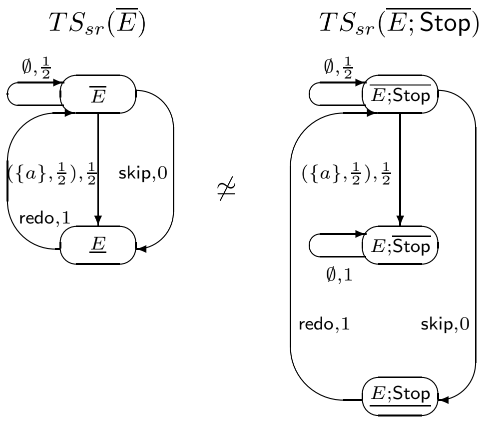

Figure 23: The sr-transition systems of $\overline{E}$ and $\overline{E;\mathsf{Stop}}$ for $E = (\{a\}, \frac{1}{2})$

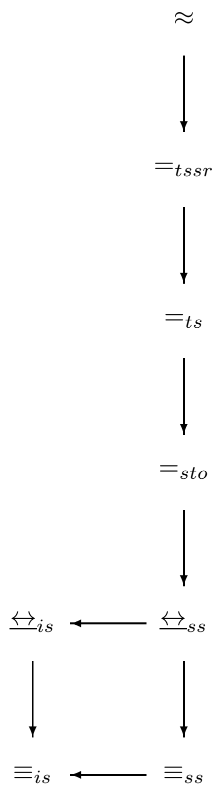

Figure 24: Interrelations of the stochastic equivalences and the new congruence

Theorem 9.1 *Let* $\leftrightarrow, \leftrightarrow\!\!\!\leftrightarrow \in \{\equiv, \underline{\leftrightarrow}, =, \approx\}$ *and* $\star, \star\star \in \{\_, is, ss, sto, ts, tssr\}$. *For dynamic expressions* G *and* G'

$$G \leftrightarrow_{\star} G' \;\Rightarrow\; G \leftrightarrow\!\!\!\leftrightarrow_{\star\star} G'$$

iff there exists a directed path from $\leftrightarrow_{\star}$ *to* $\leftrightarrow\!\!\!\leftrightarrow_{\star\star}$ *in the graph in Figure 24.*

Proof. ($\Leftarrow$) Let us check the validity of implications in the graph in Figure 24.

- The implication $=_{tssr} \rightarrow =_{ts}$ is valid, since sr-transition systems have more states and transitions than usual ones.
- The implication $\approx \rightarrow =_{tssr}$ is valid, since the sr-transition system of a dynamic formula is defined based on its structural equivalence class.

($\Rightarrow$) The absence of additional nontrivial arrows (not resulting from the combination of the existing ones by transitivity) in the graph in Figure 24 is proved by the following examples.

- Let $E = (\{a\}, \frac{1}{2})$ and $E' = (\{a\}, \frac{1}{2}); \mathsf{Stop}$. We have $\overline{E} =_{ts} \overline{E'}$ as demonstrated in the example from the proof of Proposition 9.1. On the other hand, $\overline{E} \neq_{tssr} \overline{E'}$, since only in $TS_{sr}(\overline{E'})$ after the transition with multiaction part of label $\{a\}$ we do not reach the final state (see Example 9.2).
- Let $E = (\{a\}, \frac{1}{2})$ and $E' = ((\{a\}, \frac{1}{2}); (\{\hat{a}\}, \frac{1}{2}))\ \mathsf{sy}\ a$. Then $\overline{E} =_{tssr} \overline{E'}$, since $\overline{E} =_{ts} \overline{E'}$ as demonstrated in the last example from the proof of Theorem 5.2, and the final states of both $TS_{sr}(\overline{E'})$ and $TS_{sr}(\overline{E'})$ are reachable from the others with "normal" transitions (i.e. not with skip only). On the other hand, $\overline{E} \not\approx \overline{E'}$.

□

The following theorem demonstrates that $=_{tssr}$ is a congruence of static expressions with respect to the operations of $dtsPBC$.

Theorem 9.2 *Let* $a \in Act$ *and* $E, E', F, K \in RegStatExpr$. *If* $\overline{E} =_{tssr} \overline{E'}$ *then*

1. $\overline{E \circ F} =_{tssr} \overline{E' \circ F},\ \overline{F \circ E} =_{tssr} \overline{F \circ E'},\ \circ \in \{;, [], \|\}$;
2. $\overline{E[f]} =_{tssr} \overline{E'[f]}$;
3. $\overline{E \circ a} =_{tssr} \overline{E' \circ a},\ \circ \in \{\mathsf{rs}, \mathsf{sy}\}$;
4. $\overline{[E * F * K]} =_{tssr} \overline{[E' * F * K]},\ \overline{[F * E * K]} =_{tssr} \overline{[F * E' * K]},\ \overline{[F * K * E]} =_{tssr} \overline{[F * K * E']}$.

Proof. First, we have no problems with termination, hence, the composite sr-transition systems built from the isomorphic ones can always execute the same multisets of activities. Second, the probabilities of the corresponding transitions of the composite systems coincide, since the probabilities are calculated from identical values. □

10 Performance evaluation

The standard analysis technique for DTMCs consists in the investigation of their transient and stationary behaviour and the subsequent calculation of some performance indices based on the steady-state probabilities. In this section with a case studies of a number of systems we demonstrate how steady-state distribution can be used for performance evaluation. The examples also illustrate the method of performance analysis simplification described above. The behaviour of all the systems which we consider here includes non-empty transitions only.

10.1 Shared memory system

10.1.1 The standard system

Consider a model of two processors accessing a common shared memory described in [6, 7, 92] in the continuous time setting on GSPNs. We shall analyze this shared memory system in the discrete time stochastic setting of $dtsPBC$, where concurrent execution of activities is possible, while no two transitions of a GSPN may fire simultaneously (in parallel). The model works as follows. After activation of the system (turning the computer on), two processors are active, and the common memory is available. Each processor can request an access to the memory. When a processor starts acquisition of the memory, the other processor should wait until the

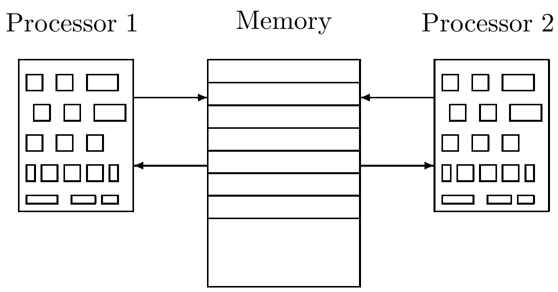

Figure 25: The diagram of the shared memory system

former one ends its memory operations, and the system returns to the state with both active processors and the available common memory. The diagram of the system is depicted in Figure 25.

Let us explain the meaning of actions from syntax of the *dtsPBC* expressions which will specify the system modules. The action a corresponds to the system activation. The actions r_i ($1 \leq i \leq 2$) represent the common memory request of processor i. The actions b_i and e_i correspond to the beginning and the end, respectively, of the common memory access of processor i. The other actions are used for communication purposes only via synchronization, and we abstract from them later using restriction.

The static expression of the first processor is $E_1 = [(\{x_1\}, \frac{1}{2}) * ((\{r_1\}, \frac{1}{2}); (\{b_1, y_1\}, \frac{1}{2}); (\{e_1, z_1\}, \frac{1}{2})) * \mathsf{Stop}]$. The static expression of the second processor is $E_2 = [(\{x_2\}, \frac{1}{2}) * ((\{r_2\}, \frac{1}{2}); (\{b_2, y_2\}, \frac{1}{2}); (\{e_2, z_2\}, \frac{1}{2})) * \mathsf{Stop}]$. The static expression of the shared memory is $E_3 = [(\{a, \widehat{x_1}, \widehat{x_2}\}, \frac{1}{2}) * (((\{\widehat{y_1}\}, \frac{1}{2}); (\{\widehat{z_1}\}, \frac{1}{2}))[]((\{\widehat{y_2}\}, \frac{1}{2}); (\{\widehat{z_2}\}, \frac{1}{2}))) * \mathsf{Stop}]$. The static expression of the shared memory system with two processors is $E = (E_1 \| E_2 \| E_3)$ $\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2$.

Let us illustrate an effect of synchronization. In the result of synchronization of activities $(\{b_i, y_i\}, \frac{1}{2})$ and $(\{\widehat{y_i}\}, \frac{1}{2})$ we obtain the new activity $(\{b_i\}, \frac{1}{4})$ $(1 \leq i \leq 2)$. The synchronization of $(\{e_i, z_i\}, \frac{1}{2})$ and $(\{\widehat{z_i}\}, \frac{1}{2})$ produces $(\{e_i\}, \frac{1}{4})$ $(1 \leq i \leq 2)$. The result of synchronization of $(\{a, \widehat{x_1}, \widehat{x_2}\}, \frac{1}{2})$ with $(\{x_1\}, \frac{1}{2})$ is $(\{a, \widehat{x_2}\}, \frac{1}{4})$, and that of synchronization of $(\{a, \widehat{x_1}, \widehat{x_2}\}, \frac{1}{2})$ with $(\{x_2\}, \frac{1}{2})$ is $(\{a, \widehat{x_1}\}, \frac{1}{4})$. After applying synchronization to $(\{a, \widehat{x_2}\}, \frac{1}{4})$ and $(\{x_2\}, \frac{1}{2})$, as well as to $(\{a, \widehat{x_1}\}, \frac{1}{4})$ and $(\{x_1\}, \frac{1}{2})$, we obtain the same activity $(\{a\}, \frac{1}{8})$.

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

$$\begin{array}{c}
s_1 = [([\overline{(\{x_1\}, \frac{1}{2})} * ((\{r_1\}, \frac{1}{2}); (\{b_1, y_1\}, \frac{1}{2}); (\{e_1, z_1\}, \frac{1}{2})) * \mathsf{Stop}] \| \\
[\overline{(\{x_2\}, \frac{1}{2})} * ((\{r_2\}, \frac{1}{2}); (\{b_2, y_2\}, \frac{1}{2}); (\{e_2, z_2\}, \frac{1}{2})) * \mathsf{Stop}] \| \\
[\overline{(\{a, \widehat{x_1}, \widehat{x_2}\}, \frac{1}{2})} * (((\{\widehat{y_1}\}, \frac{1}{2}); (\{\widehat{z_1}\}, \frac{1}{2}))[]((\{\widehat{y_2}\}, \frac{1}{2}); (\{\widehat{z_2}\}, \frac{1}{2}))) * \mathsf{Stop}]) \\
\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2]_{\approx},
\end{array}$$

$$\begin{array}{c}
s_2 = [([(\{x_1\}, \frac{1}{2}) * (\overline{(\{r_1\}, \frac{1}{2})}; (\{b_1, y_1\}, \frac{1}{2}); (\{e_1, z_1\}, \frac{1}{2})) * \mathsf{Stop}] \| \\
[(\{x_2\}, \frac{1}{2}) * (\overline{(\{r_2\}, \frac{1}{2})}; (\{b_2, y_2\}, \frac{1}{2}); (\{e_2, z_2\}, \frac{1}{2})) * \mathsf{Stop}] \| \\
[(\{a, \widehat{x_1}, \widehat{x_2}\}, \frac{1}{2}) * \overline{(((\{\widehat{y_1}\}, \frac{1}{2}); (\{\widehat{z_1}\}, \frac{1}{2}))[]((\{\widehat{y_2}\}, \frac{1}{2}); (\{\widehat{z_2}\}, \frac{1}{2})))} * \mathsf{Stop}]) \\
\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2]_{\approx},
\end{array}$$

$$\begin{array}{c}
s_3 = [([(\{x_1\}, \frac{1}{2}) * ((\{r_1\}, \frac{1}{2}); \overline{(\{b_1, y_1\}, \frac{1}{2})}; (\{e_1, z_1\}, \frac{1}{2})) * \mathsf{Stop}] \| \\
[(\{x_2\}, \frac{1}{2}) * (\overline{(\{r_2\}, \frac{1}{2})}; (\{b_2, y_2\}, \frac{1}{2}); (\{e_2, z_2\}, \frac{1}{2})) * \mathsf{Stop}] \| \\
[(\{a, \widehat{x_1}, \widehat{x_2}\}, \frac{1}{2}) * (((\overline{(\{\widehat{y_1}\}, \frac{1}{2})}; (\{\widehat{z_1}\}, \frac{1}{2}))[]((\{\widehat{y_2}\}, \frac{1}{2}); (\{\widehat{z_2}\}, \frac{1}{2}))) * \mathsf{Stop}]) \\
\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2]_{\approx},
\end{array}$$

$$\begin{array}{c}
s_4 = [([(\{x_1\}, \frac{1}{2}) * (\overline{(\{r_1\}, \frac{1}{2})}; (\{b_1, y_1\}, \frac{1}{2}); (\{e_1, z_1\}, \frac{1}{2})) * \mathsf{Stop}] \| \\
[(\{x_2\}, \frac{1}{2}) * ((\{r_2\}, \frac{1}{2}); \overline{(\{b_2, y_2\}, \frac{1}{2})}; (\{e_2, z_2\}, \frac{1}{2})) * \mathsf{Stop}] \| \\
[(\{a, \widehat{x_1}, \widehat{x_2}\}, \frac{1}{2}) * (((\{\widehat{y_1}\}, \frac{1}{2}); (\{\widehat{z_1}\}, \frac{1}{2}))[](\overline{(\{\widehat{y_2}\}, \frac{1}{2})}; (\{\widehat{z_2}\}, \frac{1}{2}))) * \mathsf{Stop}]) \\
\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2]_{\approx},
\end{array}$$

$$\begin{array}{c}
s_5 = [([(\{x_1\}, \frac{1}{2}) * ((\{r_1\}, \frac{1}{2}); (\{b_1, y_1\}, \frac{1}{2}); \overline{(\{e_1, z_1\}, \frac{1}{2})}) * \mathsf{Stop}] \| \\
[(\{x_2\}, \frac{1}{2}) * (\overline{(\{r_2\}, \frac{1}{2})}; (\{b_2, y_2\}, \frac{1}{2}); (\{e_2, z_2\}, \frac{1}{2})) * \mathsf{Stop}] \| \\
[(\{a, \widehat{x_1}, \widehat{x_2}\}, \frac{1}{2}) * (((\{\widehat{y_1}\}, \frac{1}{2}); \overline{(\{\widehat{z_1}\}, \frac{1}{2})})[]((\{\widehat{y_2}\}, \frac{1}{2}); (\{\widehat{z_2}\}, \frac{1}{2}))) * \mathsf{Stop}]) \\
\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2]_{\approx},
\end{array}$$

$TS^*(\overline{E})$

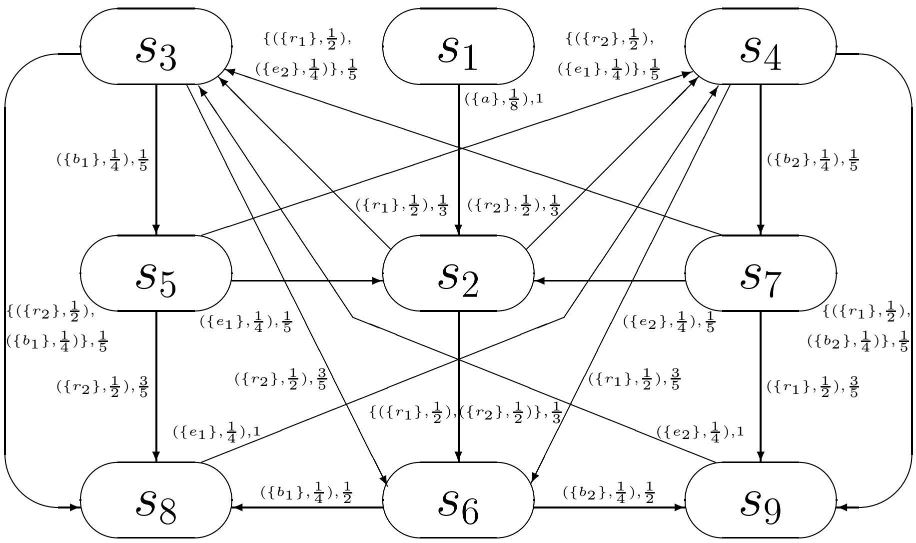

Figure 26: The transition system without empty loops of the shared memory system

$$
\begin{aligned}
s_6 = [([(\{x_1\},\tfrac{1}{2}) * ((\{r_1\},\tfrac{1}{2});\overline{(\{b_1,y_1\},\tfrac{1}{2})};(\{e_1,z_1\},\tfrac{1}{2})) * \mathsf{Stop}]\| \\
[(\{x_2\},\tfrac{1}{2}) * \underline{((\{r_2\},\tfrac{1}{2});\overline{(\{b_2,y_2\},\tfrac{1}{2})};(\{e_2,z_2\},\tfrac{1}{2}))} * \mathsf{Stop}]\| \\
[(\{a,\widehat{x_1},\widehat{x_2}\},\tfrac{1}{2}) * \overline{(((\{\widehat{y_1}\},\tfrac{1}{2});(\{\widehat{z_1}\},\tfrac{1}{2}))[](((\{\widehat{y_2}\},\tfrac{1}{2});(\{\widehat{z_2}\},\tfrac{1}{2})))} * \mathsf{Stop}]) \\
\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2]_{\approx},
\end{aligned}
$$

$$
\begin{aligned}
s_7 = [([(\{x_1\},\tfrac{1}{2}) * (\overline{(\{r_1\},\tfrac{1}{2})};(\{b_1,y_1\},\tfrac{1}{2});(\{e_1,z_1\},\tfrac{1}{2})) * \mathsf{Stop}]\| \\
[(\{x_2\},\tfrac{1}{2}) * ((\{r_2\},\tfrac{1}{2});(\{b_2,y_2\},\tfrac{1}{2});\overline{(\{e_2,z_2\},\tfrac{1}{2})}) * \mathsf{Stop}]\| \\
[(\{a,\widehat{x_1},\widehat{x_2}\},\tfrac{1}{2}) * (((\{\widehat{y_1}\},\tfrac{1}{2});(\{\widehat{z_1}\},\tfrac{1}{2}))[](((\{\widehat{y_2}\},\tfrac{1}{2});\overline{(\{\widehat{z_2}\},\tfrac{1}{2})})) * \mathsf{Stop}]) \\
\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2]_{\approx},
\end{aligned}
$$

$$
\begin{aligned}
s_8 = [([(\{x_1\},\tfrac{1}{2}) * ((\{r_1\},\tfrac{1}{2});(\{b_1,y_1\},\tfrac{1}{2});\overline{(\{e_1,z_1\},\tfrac{1}{2})}) * \mathsf{Stop}]\| \\
[(\{x_2\},\tfrac{1}{2}) * ((\{r_2\},\tfrac{1}{2});\underline{(\{b_2,y_2\},\tfrac{1}{2})};(\{e_2,z_2\},\tfrac{1}{2})) * \mathsf{Stop}]\| \\
[(\{a,\widehat{x_1},\widehat{x_2}\},\tfrac{1}{2}) * (((\{\widehat{y_1}\},\tfrac{1}{2});\overline{(\{\widehat{z_1}\},\tfrac{1}{2})})[](((\{\widehat{y_2}\},\tfrac{1}{2});(\{\widehat{z_2}\},\tfrac{1}{2}))) * \mathsf{Stop}]) \\
\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2]_{\approx},
\end{aligned}
$$

$$
\begin{aligned}
s_9 = [([(\{x_1\},\tfrac{1}{2}) * ((\{r_1\},\tfrac{1}{2});\overline{(\{b_1,y_1\},\tfrac{1}{2})};(\{e_1,z_1\},\tfrac{1}{2})) * \mathsf{Stop}]\| \\
[(\{x_2\},\tfrac{1}{2}) * ((\{r_2\},\tfrac{1}{2});(\{b_2,y_2\},\tfrac{1}{2});\overline{(\{e_2,z_2\},\tfrac{1}{2})}) * \mathsf{Stop}]\| \\
[(\{a,\widehat{x_1},\widehat{x_2}\},\tfrac{1}{2}) * (((\{\widehat{y_1}\},\tfrac{1}{2});(\{\widehat{z_1}\},\tfrac{1}{2}))[](((\{\widehat{y_2}\},\tfrac{1}{2});\overline{(\{\widehat{z_2}\},\tfrac{1}{2})})) * \mathsf{Stop}]) \\
\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2]_{\approx}.
\end{aligned}
$$

The states are interpreted as follows: s_1 is the initial state, s_2: the system is activated and the memory is not requested, s_3: the memory is requested by the first processor, s_4: the memory is requested by the second processor, s_5: the memory is allocated to the first processor, s_6: the memory is requested by two processors, s_7: the memory is allocated to the second processor, s_8: the memory is allocated to the first processor and the memory is requested by the second processor, s_9: the memory is allocated to the second processor and the memory is requested by the first processor.

In Figure 26, the transition system without empty loops $TS^*(\overline{E})$ is presented. In Figure 27, the underlying DTMC without empty loops $DTMC^*(\overline{E})$ is depicted.

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

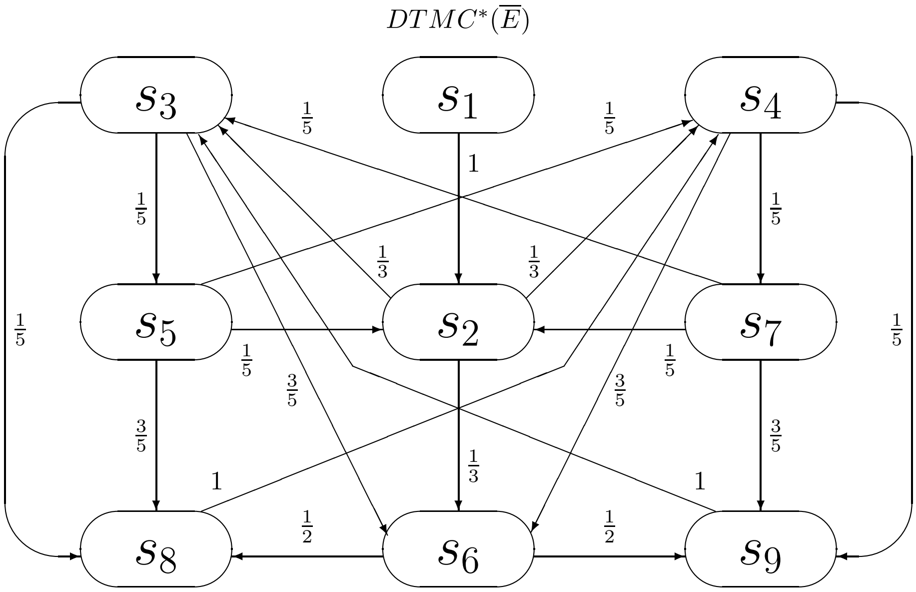

Figure 27: The underlying DTMC without empty loops of the shared memory system

Table 8: Transient and steady-state probabilities of the shared memory system

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

$$\mathbf{P}^* = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{5} & \frac{3}{5} & 0 & \frac{1}{5} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{3}{5} & \frac{1}{5} & 0 & \frac{1}{5} \\ 0 & \frac{1}{5} & 0 & \frac{1}{5} & 0 & 0 & 0 & \frac{3}{5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{5} & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 & \frac{3}{5} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

In Table 8, the transient and the steady-state probabilities $\psi_i^*[k]$ $(i \in \{1, 2, 3, 5, 6, 8\})$ of the shared memory system at the time moments k $(0 \le k \le 10)$ and $k = \infty$ are presented, and in Figure 28, the alteration diagram (evolution in time) for the transient probabilities is depicted. It is sufficient to consider the probabilities for the states $s_1, s_2, s_3, s_5, s_6, s_8$ only, since the corresponding values coincide for s_3, s_4, as well as for s_5, s_7, and for s_8, s_9.

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

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

We can now calculate the main performance indices.

- The average recurrence time in the state s_2, where no processor requests the memory, called the *average system run-through*, is $\frac{1}{\psi_2^*} = \frac{209}{3} = 69\frac{2}{3}$.

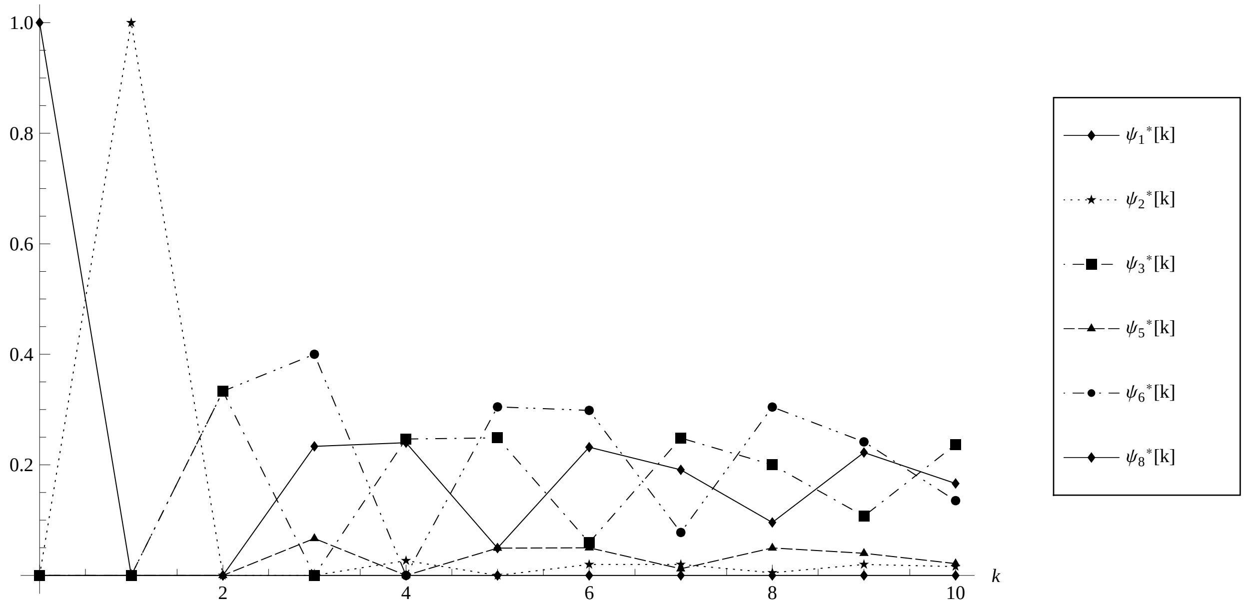

Figure 28: Transient probabilities alteration diagram of the shared memory system

- The common memory is available only in the states s_2, s_3, s_4, s_6. The steady-state probability that the memory is available is $\psi_2^* + \psi_3^* + \psi_4^* + \psi_6^* = \frac{3}{209} + \frac{75}{418} + \frac{75}{418} + \frac{46}{209} = \frac{124}{209}$. Then the steady-state probability that the memory is used (i.e. not available), called the *shared memory utilization*, is $1 - \frac{124}{209} = \frac{85}{209}$.
- The common memory request of the first processor $(\{r_1\}, \frac{1}{2})$ is only possible from the states s_2, s_4, s_7. In each of the states, the request probability is the sum of the execution probabilities for all multisets of activities containing $(\{r_1\}, \frac{1}{2})$. Thus, the *steady-state probability of the shared memory request from the first processor* is $\psi_2^* \sum_{\{\Gamma|(\{r_1\},\frac{1}{2})\in\Gamma\}} PT^*(\Gamma, s_2) + \psi_4^* \sum_{\{\Gamma|(\{r_1\},\frac{1}{2})\in\Gamma\}} PT^*(\Gamma, s_4) +$ $\psi_7^* \sum_{\{\Gamma|(\{r_1\},\frac{1}{2})\in\Gamma\}} PT^*(\Gamma, s_7) = \frac{3}{209}\left(\frac{1}{3} + \frac{1}{3}\right) + \frac{75}{418}\left(\frac{3}{5} + \frac{1}{5}\right) + \frac{15}{418}\left(\frac{3}{5} + \frac{1}{5}\right) = \frac{38}{209}$.

In Figure 29, the marked dts-boxes corresponding to the dynamic expressions of two processors and shared memory are presented, i.e. $N_i = Box_{dts}(\overline{E_i})$ $(1 \le i \le 3)$. In Figure 30, the marked dts-box corresponding to the dynamic expression of the shared memory system is depicted, i.e. $N = Box_{dts}(\overline{E})$.

10.1.2 The abstract system and its reduction

Let us consider a modification of the shared memory system with abstraction from identifiers of the processors, i.e. such that the processors are indistinguishable. For example, we can just see that a processor requires memory or the memory is allocated to it but cannot observe which processor is it. We call this system the abstract shared memory one. To implement the abstraction, we replace the actions r_i, b_i, e_i $(1 \le i \le 2)$ in the system specification by r, b, e, respectively.

The static expression of the first processor is $F_1 = [(\{x_1\}, \frac{1}{2}) * ((\{r\}, \frac{1}{2}); (\{b, y_1\}, \frac{1}{2}); (\{e, z_1\}, \frac{1}{2})) * \mathsf{Stop}]$. The static expression of the second processor is $F_2 = [(\{x_2\}, \frac{1}{2}) * ((\{r\}, \frac{1}{2}); (\{b, y_2\}, \frac{1}{2}); (\{e, z_2\}, \frac{1}{2})) * \mathsf{Stop}]$. The static expression of the shared memory is $F_3 = [(\{a, \widehat{x_1}, \widehat{x_2}\}, \frac{1}{2}) * (((\{\widehat{y_1}\}, \frac{1}{2}); (\{\widehat{z_1}\}, \frac{1}{2}))[](((\{\widehat{y_2}\}, \frac{1}{2}); (\{\widehat{z_2}\}, \frac{1}{2}))) * \mathsf{Stop}]$. The static expression of the abstract shared memory system with two processors is $F = (F_1 \| F_2 \| F_3)$ sy x_1 sy x_2 sy y_1 sy y_2 sy z_1 sy z_2 rs x_1 rs x_2 rs y_1 rs y_2 rs z_1 rs z_2.

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

The first and second performance indices are the same for the standard and the abstract systems. Let us consider the following performance index based on non-identified viewpoint to the processors.

- The common memory request of a processor $(\{r\}, \frac{1}{2})$ is only possible from the states s_2, s_3, s_4, s_5, s_7. In each of the states, the request probability is the sum of the execution probabilities for all multisets of activities containing $(\{r\}, \frac{1}{2})$. Thus, the *steady-state probability of the shared memory request from a processor* is $\psi_2^* \sum_{\{\Gamma|(\{r\},\frac{1}{2})\in\Gamma\}} PT^*(\Gamma, s_2) + \psi_3^* \sum_{\{\Gamma|(\{r\},\frac{1}{2})\in\Gamma\}} PT^*(\Gamma, s_3) + \psi_4^* \sum_{\{\Gamma|(\{r\},\frac{1}{2})\in\Gamma\}} PT^*(\Gamma, s_4) +$

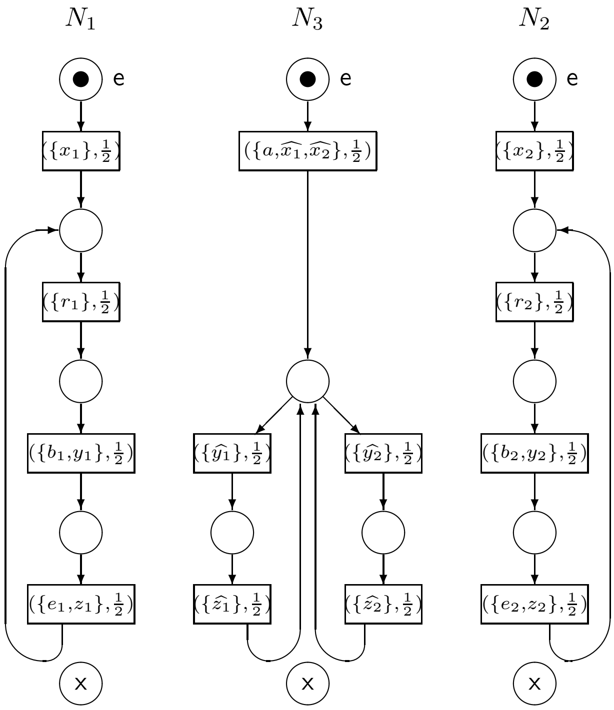

Figure 29: The marked dts-boxes of two processors and shared memory

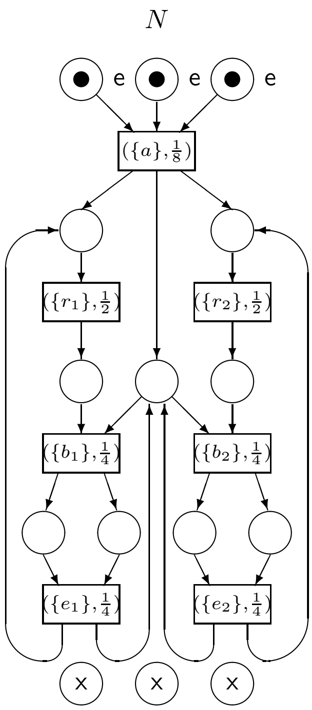

Figure 30: The marked dts-box of the shared memory system

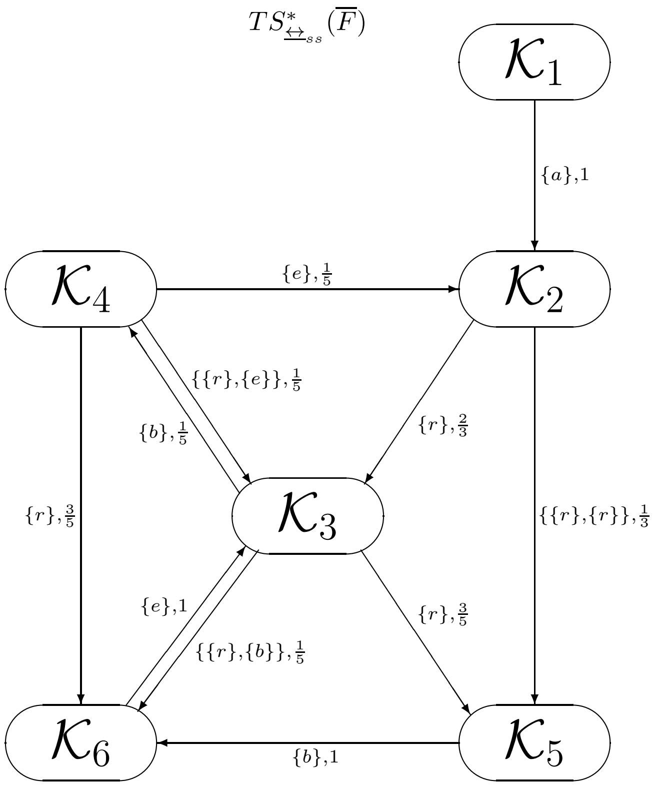

Figure 31: The quotient transition system without empty loops of the abstract shared memory system

$\psi_5^* \sum_{\{\Gamma|(\{r\},\frac{1}{2})\in\Gamma\}} PT^*(\Gamma, s_5) + \psi_7^* \sum_{\{\Gamma|(\{r\},\frac{1}{2})\in\Gamma\}} PT^*(\Gamma, s_7) = \frac{3}{209}\left(\frac{1}{3}+\frac{1}{3}+\frac{1}{3}\right) + \frac{75}{418}\left(\frac{3}{5}+\frac{1}{5}\right) + \frac{75}{418}\left(\frac{3}{5}+\frac{1}{5}\right) + \frac{15}{418}\left(\frac{3}{5}+\frac{1}{5}\right) + \frac{15}{418}\left(\frac{3}{5}+\frac{1}{5}\right) = \frac{75}{209}$.

The marked dts-boxes corresponding to the dynamic expressions of the standard and the abstract two processors and shared memory are similar, as well as the marked dts-boxes corresponding to the dynamic expression of the standard and the abstract shared memory systems.

Let us consider a reduction of the abstract shared memory system. Note that $TS^*(\overline{F})$ can be reduced by merging the equivalent states s_3, s_4, as well as s_5, s_7, as well as s_8, s_9, thus, it can be transformed into a transition system with six states only. But the resulting reduction of the initial transition system $TS^*(\overline{F})$ will not correspond to some $dtsPBC$ expression anymore.

We have $DR(\overline{F})/_{\mathcal{R}_{ss}(\overline{F})} = \{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_4, \mathcal{K}_5, \mathcal{K}_6\}$, where $\mathcal{K}_1 = \{s_1\}$ (the initial state), $\mathcal{K}_2 = \{s_2\}$ (the system is activated and the memory is not requested), $\mathcal{K}_3 = \{s_3, s_4\}$ (the memory is requested by one processor), $\mathcal{K}_4 = \{s_5, s_7\}$ (the memory is allocated to a processor), $\mathcal{K}_5 = \{s_6\}$ (the memory is requested by two processors), $\mathcal{K}_6 = \{s_8, s_9\}$ (the memory is allocated to a processor and the memory is requested by another processor).

In Figure 31, the quotient transition system without empty loops $TS^*_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is presented. In Figure 32, the quotient underlying DTMC without empty loops $DTMC^*_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is depicted.

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

$$\mathbf{P}'^* = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{3} & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{1}{5} & \frac{3}{5} & \frac{1}{5} \\ 0 & \frac{1}{5} & \frac{1}{5} & 0 & 0 & \frac{3}{5} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

In Table 9, the transient and the steady-state probabilities $\psi_i'^*[k]$ $(1 \le i \le 6)$ of the quotient abstract shared memory system at the time moments k $(0 \le k \le 10)$ and $k = \infty$ are presented, and in Figure 33, the alteration diagram (evolution in time) for the transient probabilities is depicted.

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

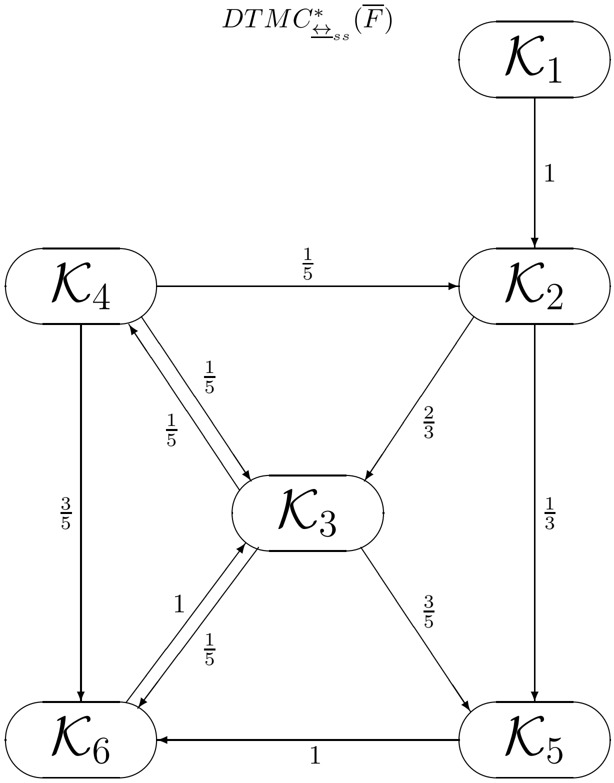

Figure 32: The quotient underlying DTMC without empty loops of the abstract shared memory system

Table 9: Transient and steady-state probabilities of the quotient abstract shared memory system

k	0	1	2	3	4	5	6	7	8	9	10	∞
${\psi'_1}^*[k]$	1	0	0	0	0	0	0	0	0	0	0	0
${\psi'_2}^*[k]$	0	1	0	0	0.0267	0	0.0197	0.0199	0.0047	0.0199	0.0160	0.0144
${\psi'_3}^*[k]$	0	0	0.6667	0	0.4933	0.4978	0.1184	0.4967	0.4001	0.2142	0.4735	0.3589
${\psi'_4}^*[k]$	0	0	0	0.1333	0	0.0987	0.0996	0.0237	0.0993	0.0800	0.0428	0.0718
${\psi'_5}^*[k]$	0	0	0.3333	0.4000	0	0.3049	0.2987	0.0776	0.3047	0.2416	0.1351	0.2201
${\psi'_6}^*[k]$	0	0	0	0.4667	0.4800	0.0987	0.4636	0.3821	0.1912	0.4443	0.3325	0.3349

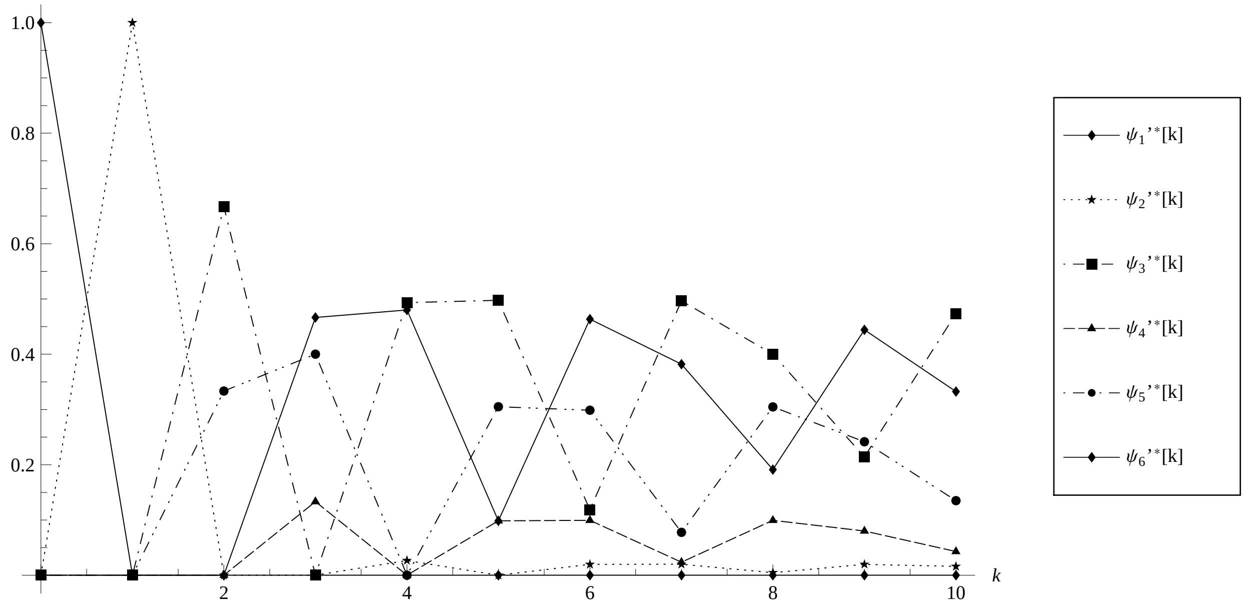

Figure 33: Transient probabilities alteration diagram of the quotient abstract shared memory system

$$\psi'^{*} = \left(0, \frac{3}{209}, \frac{75}{209}, \frac{15}{209}, \frac{46}{209}, \frac{70}{209}\right).$$

We can now calculate the main performance indices.

- The average recurrence time in the state $\mathcal{K}_2$, where no processor requests the memory, called the *average system run-through*, is $\frac{1}{\psi_2'^{*}} = \frac{209}{3} = 69\frac{2}{3}$.
- The common memory is available only in the states $\mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_5$. The steady-state probability that the memory is available is $\psi_2'^{*} + \psi_3'^{*} + \psi_5'^{*} = \frac{3}{209} + \frac{75}{209} + \frac{46}{209} = \frac{124}{209}$. Then the steady-state probability that the memory is used (i.e. not available), called the *shared memory utilization*, is $1 - \frac{124}{209} = \frac{85}{209}$.
- The common memory request of a processor $\{r\}$ is only possible from the states $\mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_4$. In each of the states, the request probability is the sum of the execution probabilities for all multisets of multiactions containing $\{r\}$. Thus, the *steady-state probability of the shared memory request from a processor* is $\psi_2'^{*} \sum_{\{A,\mathcal{K}|\{r\}\in A,\ \mathcal{K}_2 \xrightarrow{A} \mathcal{K}\}} PM_A^*(\mathcal{K}_2, \mathcal{K}) + \psi_3'^{*} \sum_{\{A,\mathcal{K}|\{r\}\in A,\ \mathcal{K}_3 \xrightarrow{A} \mathcal{K}\}} PM_A^*(\mathcal{K}_3, \mathcal{K}) +$
$\psi_4'^{*} \sum_{\{A,\mathcal{K}|\{r\}\in A,\ \mathcal{K}_4 \xrightarrow{A} \mathcal{K}\}} PM_A^*(\mathcal{K}_4, \mathcal{K}) = \frac{3}{209}\left(\frac{2}{3} + \frac{1}{3}\right) + \frac{75}{209}\left(\frac{3}{5} + \frac{1}{5}\right) + \frac{15}{209}\left(\frac{3}{5} + \frac{1}{5}\right) = \frac{75}{209}$.

One can see that the performance indices are the same for the complete and the quotient abstract shared memory systems. The coincidence of the first and second performance indices obviously illustrates the result of Proposition 8.1. The coincidence of the third performance index is due to Theorem 8.2: one should just apply its result to the step traces $\{\{r\}\}$, $\{\{r\},\{r\}\}$, $\{\{r\},\{b\}\}$, $\{\{r\},\{e\}\}$ of the expression $\overline{F}$ and itself, and then sum the left and right parts of the three resulting equalities.

10.1.3 The generalized system

Let us determine which is the influence of the multiaction probabilities from specification of the shared memory system on its performance. Suppose that all the mentioned multiactions have the same generalized probability $\rho \in (0; 1)$. The resulting specification K of the generalized shared memory system is defined as follows.

The static expression of the first processor is $K_1 = [(\{x_1\}, \rho) * ((\{r_1\}, \rho); (\{b_1, y_1\}, \rho); (\{e_1, z_1\}, \rho)) * \mathsf{Stop}]$. The static expression of the second processor is $K_2 = [(\{x_2\}, \rho) * ((\{r_2\}, \rho); (\{b_2, y_2\}, \rho); (\{e_2, z_2\}, \rho)) * \mathsf{Stop}]$. The static expression of the shared memory is $K_3 = [(\{a, \widehat{x_1}, \widehat{x_2}\}, \rho) * (((\{\widehat{y_1}\}, \rho); (\{\widehat{z_1}\}, \rho))[]((\{\widehat{y_2}\}, \rho); (\{\widehat{z_2}\}, \rho))) * \mathsf{Stop}]$. The static expression of the generalized shared memory system with two processors is $K = (K_1 \| K_2 \| K_3)$ sy x_1 sy x_2 sy y_1 sy y_2 sy z_1 sy z_2 rs x_1 rs x_2 rs y_1 rs y_2 rs z_1 rs z_2.

$DR(\overline{K})$ consists of the 9 states which are interpreted as follows: $\tilde{s}_1$ is the initial state, $\tilde{s}_2$: the system is activated and the memory is not requested, $\tilde{s}_3$: the memory is requested by the first processor, $\tilde{s}_4$: the memory is requested by the second processor, $\tilde{s}_5$: the memory is allocated to the first processor, $\tilde{s}_6$: the memory is requested by two processors, $\tilde{s}_7$: the memory is allocated to the second processor, $\tilde{s}_8$: the memory is allocated to the first processor and the memory is requested by the second processor, $\tilde{s}_9$: the memory is allocated to the second processor and the memory is requested by the first processor.

The TPM for $DTMC^*(\overline{K})$ is

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

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

$$\tilde{\psi}^* = \tfrac{1}{2(6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5)}(0, 2\rho^2(2-\rho)(1-\rho)^2, (2-\rho)(1+\rho-\rho^2)^2, (2-\rho)(1+\rho-\rho^2)^2,$$
$$\rho(2-\rho-4\rho^2+4\rho^3-\rho^4), 2(2+\rho-5\rho^2+\rho^3+\rho^4), \rho(2-\rho-4\rho^2+4\rho^3-\rho^4),$$
$$2+3\rho-6\rho^2+\rho^3+\rho^4, 2+3\rho-6\rho^2+\rho^3+\rho^4).$$

We can now calculate the main performance indices.

- The average recurrence time in the state $\tilde{s}_2$, where no processor requests the memory, called the *average system run-through*, is $\frac{1}{\tilde{\psi}_2^*} = \frac{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}{\rho^2(2-\rho)(1-\rho)^2}$.

- The common memory is available only in the states $\tilde{s}_2, \tilde{s}_3, \tilde{s}_4, \tilde{s}_6$. The steady-state probability that the memory is available is $\tilde{\psi}_2^* + \tilde{\psi}_3^* + \tilde{\psi}_4^* + \tilde{\psi}_6^* = \frac{\rho^2(2-\rho)(1-\rho)^2}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5} + \frac{(2-\rho)(1+\rho-\rho^2)^2}{2(6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5)} + \frac{(2-\rho)(1+\rho-\rho^2)^2}{2(6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5)} + \frac{2+\rho-5\rho^2+\rho^3+\rho^4}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5} = \frac{4+4\rho-7\rho^2-7\rho^3+9\rho^4-2\rho^5}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}$. Then the steady-state probability that the memory is used (i.e. not available), called the *shared memory utilization*, is $1 - \frac{4+4\rho-7\rho^2-7\rho^3+9\rho^4-2\rho^5}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5} = \frac{2+5\rho-7\rho^2-3\rho^3+5\rho^4-\rho^5}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}$.

- The common memory request of the first processor $(\{r_1\}, \rho)$ is only possible from the states $\tilde{s}_2, \tilde{s}_4, \tilde{s}_7$. In each of the states, the request probability is the sum of the execution probabilities for all multisets of activities containing $(\{r_1\}, \rho)$. Thus, the *steady-state probability of the shared memory request from the first processor* is
$\tilde{\psi}_2^* \sum_{\{\Gamma|(\{r_1\},\rho)\in\Gamma\}} PT^*(\Gamma, \tilde{s}_2) + \tilde{\psi}_4^* \sum_{\{\Gamma|(\{r_1\},\rho)\in\Gamma\}} PT^*(\Gamma, \tilde{s}_4) + \tilde{\psi}_7^* \sum_{\{\Gamma|(\{r_1\},\rho)\in\Gamma\}} PT^*(\Gamma, \tilde{s}_7) = \frac{\rho^2(2-\rho)(1-\rho)^2}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}\left(\frac{1-\rho}{2-\rho} + \frac{\rho}{2-\rho}\right) + \frac{(2-\rho)(1+\rho-\rho^2)^2}{2(6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5)}\left(\frac{1-\rho^2}{1+\rho-\rho^2} + \frac{\rho^2}{1+\rho-\rho^2}\right) + \frac{\rho(2-\rho-4\rho^2+4\rho^3-\rho^4)}{2(6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5)}\left(\frac{1-\rho^2}{1+\rho-\rho^2} + \frac{\rho^2}{1+\rho-\rho^2}\right) = \frac{2+3\rho-4\rho^2-2\rho^3+2\rho^4}{2(6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5)}$.

10.1.4 The abstract generalized system and its reduction

Let us consider a modification of the generalized shared memory system with abstraction from identifiers of the processors. We call this system the abstract generalized shared memory one.

The static expression of the first processor is $L_1 = [(\{x_1\}, \rho) * ((\{r\}, \rho); (\{b, y_1\}, \rho); (\{e, z_1\}, \rho)) * \mathsf{Stop}]$. The static expression of the second processor is $L_2 = [(\{x_2\}, \rho) * ((\{r\}, \rho); (\{b, y_2\}, \rho); (\{e, z_2\}, \rho)) * \mathsf{Stop}]$. The static expression of the shared memory is $L_3 = [(\{a, \widehat{x_1}, \widehat{x_2}\}, \rho) * (((\{\widehat{y_1}\}, \rho); (\{\widehat{z_1}\}, \rho))[]((\{\widehat{y_2}\}, \rho); (\{\widehat{z_2}\}, \rho))) * \mathsf{Stop}]$. The static expression of the abstract shared memory generalized system with two processors is $L = (L_1 \| L_2 \| L_3)$ $\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2$.

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

The first and second performance indices are the same for the generalized system and its abstract modification. Let us consider the following performance index based on non-identified viewpoint to the processors.

- The common memory request of a processor $(\{r\}, \rho)$ is only possible from the states $\tilde{s}_2, \tilde{s}_3, \tilde{s}_4, \tilde{s}_5, \tilde{s}_7$. In each of the states, the request probability is the sum of the execution probabilities for all multisets of

activities containing $(\{r\},\rho)$. Thus, the *steady-state probability of the shared memory request from a processor* is $\tilde{\psi}_2^*\sum_{\{\Gamma|(\{r\},\rho)\in\Gamma\}}PT^*(\Gamma,\tilde{s}_2)+\tilde{\psi}_3^*\sum_{\{\Gamma|(\{r\},\rho)\in\Gamma\}}PT^*(\Gamma,\tilde{s}_3)+\tilde{\psi}_4^*\sum_{\{\Gamma|(\{r\},\rho)\in\Gamma\}}PT^*(\Gamma,\tilde{s}_4)+\tilde{\psi}_5^*\sum_{\{\Gamma|(\{r\},\rho)\in\Gamma\}}PT^*(\Gamma,\tilde{s}_5)+\tilde{\psi}_7^*\sum_{\{\Gamma|(\{r\},\rho)\in\Gamma\}}PT^*(\Gamma,\tilde{s}_7)=$
$\frac{\rho^2(2-\rho)(1-\rho)^2}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}\left(\frac{1-\rho}{2-\rho}+\frac{1-\rho}{2-\rho}+\frac{\rho}{2-\rho}\right)+\frac{(2-\rho)(1+\rho-\rho^2)^2}{2(6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5)}\left(\frac{1-\rho^2}{1+\rho-\rho^2}+\frac{\rho^2}{1+\rho-\rho^2}\right)+$
$\frac{(2-\rho)(1+\rho-\rho^2)^2}{2(6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5)}\left(\frac{1-\rho^2}{1+\rho-\rho^2}+\frac{\rho^2}{1+\rho-\rho^2}\right)+\frac{\rho(2-\rho-4\rho^2+4\rho^3-\rho^4)}{2(6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5)}\left(\frac{1-\rho^2}{1+\rho-\rho^2}+\frac{\rho^2}{1+\rho-\rho^2}\right)+$
$\frac{\rho(2-\rho-4\rho^2+4\rho^3-\rho^4)}{2(6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5)}\left(\frac{1-\rho^2}{1+\rho-\rho^2}+\frac{\rho^2}{1+\rho-\rho^2}\right)=\frac{(2-\rho)(1+\rho-\rho^2)^2}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}$.

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

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

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

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

$$\tilde{\psi}'^*=\frac{1}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}(0,\rho^2(2-\rho)(1-\rho)^2,(2-\rho)(1+\rho-\rho^2)^2,$$
$$\rho(2-\rho-4\rho^2+4\rho^3-\rho^4),2+\rho-5\rho^2+\rho^3+\rho^4,2+3\rho-6\rho^2+\rho^3+\rho^4).$$

We can now calculate the main performance indices.

- The average recurrence time in the state $\widetilde{\mathcal{K}}_2$, where no processor requests the memory, called the *average system run-through*, is $\frac{1}{\tilde{\psi}_2'^*}=\frac{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}{\rho^2(2-\rho)(1-\rho)^2}$.
- The common memory is available only in the states $\widetilde{\mathcal{K}}_2,\widetilde{\mathcal{K}}_3,\widetilde{\mathcal{K}}_5$. The steady-state probability that the memory is available is $\tilde{\psi}_2'^*+\tilde{\psi}_3'^*+\tilde{\psi}_5'^*=\frac{\rho^2(2-\rho)(1-\rho)^2}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}+\frac{(2-\rho)(1+\rho-\rho^2)^2}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}+\frac{2+\rho-5\rho^2+\rho^3+\rho^4}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}=\frac{4+4\rho-7\rho^2-7\rho^3+9\rho^4-2\rho^5}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}$. Then the steady-state probability that the memory is used (i.e. not available), called the *shared memory utilization*, is $1-\frac{4+4\rho-7\rho^2-7\rho^3+9\rho^4-2\rho^5}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}=\frac{2+5\rho-7\rho^2-3\rho^3+5\rho^4-\rho^5}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}$.
- The common memory request of a processor $\{r\}$ is only possible from the states $\widetilde{\mathcal{K}}_2,\widetilde{\mathcal{K}}_3,\widetilde{\mathcal{K}}_4$. In each of the states, the request probability is the sum of the execution probabilities for all multisets of multiactions containing $\{r\}$. Thus, the *steady-state probability of the shared memory request from a processor* is $\tilde{\psi}_2'^*\sum_{\{A,\widetilde{\mathcal{K}}|\{r\}\in A,\ \widetilde{\mathcal{K}}_2\xrightarrow{A}\widetilde{\mathcal{K}}\}}PM_A^*(\widetilde{\mathcal{K}}_2,\widetilde{\mathcal{K}})+\tilde{\psi}_3'^*\sum_{\{A,\widetilde{\mathcal{K}}|\{r\}\in A,\ \widetilde{\mathcal{K}}_3\xrightarrow{A}\widetilde{\mathcal{K}}\}}PM_A^*(\widetilde{\mathcal{K}}_3,\widetilde{\mathcal{K}})+$
$\tilde{\psi}_4'^*\sum_{\{A,\widetilde{\mathcal{K}}|\{r\}\in A,\ \widetilde{\mathcal{K}}_4\xrightarrow{A}\widetilde{\mathcal{K}}\}}PM_A^*(\widetilde{\mathcal{K}}_4,\widetilde{\mathcal{K}})=\frac{\rho^2(2-\rho)(1-\rho)^2}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}\left(\frac{2(1-\rho)}{2-\rho}+\frac{\rho}{2-\rho}\right)+$
$\frac{(2-\rho)(1+\rho-\rho^2)^2}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}\left(\frac{1-\rho^2}{1+\rho-\rho^2}+\frac{\rho^2}{1+\rho-\rho^2}\right)+\frac{\rho(2-\rho-4\rho^2+4\rho^3-\rho^4)}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}\left(\frac{1-\rho^2}{1+\rho-\rho^2}+\frac{\rho^2}{1+\rho-\rho^2}\right)=$
$\frac{(2-\rho)(1+\rho-\rho^2)^2}{6+9\rho-14\rho^2-10\rho^3+14\rho^4-3\rho^5}$.

One can see that the performance indices are the same for the complete and the quotient abstract generalized shared memory systems. The coincidence of the first and second performance indices obviously illustrates the result of Proposition 8.1. The coincidence of the third performance index is due to Theorem 8.2: one should just apply its result to the step traces $\{\{r\}\}$, $\{\{r\},\{r\}\}$, $\{\{r\},\{b\}\}$, $\{\{r\},\{e\}\}$ of the expression $\overline{L}$ and itself, and then sum the left and right parts of the three resulting equalities.

10.2 Dining philosophers system

10.2.1 The standard system

Consider a model of five dining philosophers, for which the Petri net interpretation was proposed in [121]. We shall investigate this dining philosophers system in the discrete time stochastic setting of $dtsPBC$, where

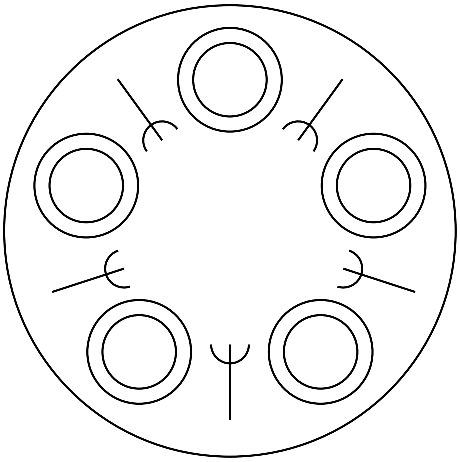

Figure 34: The diagram of the dining philosophers system

concurrent execution of activities is possible, while in the previous models of the system, based on PNs, no parallel transition firings were considered. The philosophers occupy a round table, and there is one fork between every neighboring persons, hence, there are five forks on the table. A philosopher needs two forks to eat, namely, his left and right ones. Hence, all five philosophers cannot eat together, since otherwise there will not be enough forks available, but only one of two of them who are not neighbors. The model works as follows. After the activation of the system (the philosophers come in the dining room), five forks are placed on the table. If the left and right forks are available for a philosopher, he takes them simultaneously and begins eating. At the end of eating, the philosopher places both his forks simultaneously back on the table. The strategy to pick up and release two forks simultaneously prevents the situation when a philosopher takes one fork but is not able to pick up the second one since their neighbor has already done so. In particular, we avoid a deadlock when all the philosophers take their left (right) forks and wait until their right (left) forks will be available. The diagram of the system is depicted in Figure 34.

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

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

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

The static expression of the philosopher i $(1 \le i \le 4)$ is $E_i = [(\{x_i\}, \frac{1}{2}) * (((\{b_i, \widehat{y_i}\}, \frac{1}{2}); (\{e_i, \widehat{z_i}\}, \frac{1}{2}))[] ((\{y_{i+1}\}, \frac{1}{2}); (\{z_{i+1}\}, \frac{1}{2}))) * \mathsf{Stop}]$. The static expression of the philosopher 5 is $E_5 = [(\{a, \widehat{x_1}, \widehat{x_2}, \widehat{x_2}, \widehat{x_4}\}, \frac{1}{2}) * (((\{b_5, \widehat{y_5}\}, \frac{1}{2}); (\{e_5, \widehat{z_5}\}, \frac{1}{2}))[]((\{y_1\}, \frac{1}{2}); (\{z_1\}, \frac{1}{2}))) * \mathsf{Stop}]$. The static expression of the dining philosophers system is $E = (E_1 \| E_2 \| E_3 \| E_4 \| E_5)$ sy x_1 sy x_2 sy x_3 sy x_4 sy y_1 sy y_2 sy y_3 sy y_4 sy y_5 sy z_1 sy z_2 sy z_3 sy z_4 sy z_5 rs x_1 rs x_2 rs x_3 rs x_4 rs y_1 rs y_2 rs y_3 rs y_4 rs y_5 rs z_1 rs z_2 rs z_3 rs z_4 rs z_5.

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

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

$$\begin{array}{c} s_1 = [([(\{x_1\},\frac{1}{2}) * (((\{b_1,\widehat{y_1}\},\frac{1}{2});(\{e_1,\widehat{z_1}\},\frac{1}{2}))[]((\{y_2\},\frac{1}{2});(\{z_2\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_2\},\frac{1}{2}) * (((\{b_2,\widehat{y_2}\},\frac{1}{2});(\{e_2,\widehat{z_2}\},\frac{1}{2}))[]((\{y_3\},\frac{1}{2});(\{z_3\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_3\},\frac{1}{2}) * (((\{b_3,\widehat{y_3}\},\frac{1}{2});(\{e_3,\widehat{z_3}\},\frac{1}{2}))[]((\{y_4\},\frac{1}{2});(\{z_4\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_4\},\frac{1}{2}) * (((\{b_4,\widehat{y_4}\},\frac{1}{2});(\{e_4,\widehat{z_4}\},\frac{1}{2}))[]((\{y_5\},\frac{1}{2});(\{z_5\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\frac{1}{2}) * (((\{b_5,\widehat{y_5}\},\frac{1}{2});(\{e_5,\widehat{z_5}\},\frac{1}{2}))[]((\{y_1\},\frac{1}{2});(\{z_1\},\frac{1}{2}))) * \mathsf{Stop}]) \\ \mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ x_3\ \mathsf{sy}\ x_4\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ y_3\ \mathsf{sy}\ y_4\ \mathsf{sy}\ y_5\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{sy}\ z_3\ \mathsf{sy}\ z_4\ \mathsf{sy}\ z_5 \\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ x_3\ \mathsf{rs}\ x_4\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ y_3\ \mathsf{rs}\ y_4\ \mathsf{rs}\ y_5\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2\ \mathsf{rs}\ z_3\ \mathsf{rs}\ z_4\ \mathsf{rs}\ z_5]_{\approx}, \end{array}$$

$$\begin{array}{c} s_2 = [([(\{x_1\},\frac{1}{2}) * (((\{b_1,\widehat{y_1}\},\frac{1}{2});(\{e_1,\widehat{z_1}\},\frac{1}{2}))[]((\{y_2\},\frac{1}{2});(\{z_2\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_2\},\frac{1}{2}) * (((\{b_2,\widehat{y_2}\},\frac{1}{2});(\{e_2,\widehat{z_2}\},\frac{1}{2}))[]((\{y_3\},\frac{1}{2});(\{z_3\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_3\},\frac{1}{2}) * (((\{b_3,\widehat{y_3}\},\frac{1}{2});(\{e_3,\widehat{z_3}\},\frac{1}{2}))[]((\{y_4\},\frac{1}{2});(\{z_4\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_4\},\frac{1}{2}) * (((\{b_4,\widehat{y_4}\},\frac{1}{2});(\{e_4,\widehat{z_4}\},\frac{1}{2}))[]((\{y_5\},\frac{1}{2});(\{z_5\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\frac{1}{2}) * (((\{b_5,\widehat{y_5}\},\frac{1}{2});(\{e_5,\widehat{z_5}\},\frac{1}{2}))[]((\{y_1\},\frac{1}{2});(\{z_1\},\frac{1}{2}))) * \mathsf{Stop}]) \\ \mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ x_3\ \mathsf{sy}\ x_4\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ y_3\ \mathsf{sy}\ y_4\ \mathsf{sy}\ y_5\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{sy}\ z_3\ \mathsf{sy}\ z_4\ \mathsf{sy}\ z_5 \\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ x_3\ \mathsf{rs}\ x_4\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ y_3\ \mathsf{rs}\ y_4\ \mathsf{rs}\ y_5\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2\ \mathsf{rs}\ z_3\ \mathsf{rs}\ z_4\ \mathsf{rs}\ z_5]_{\approx}, \end{array}$$

$$\begin{array}{c} s_3 = [([(\{x_1\},\frac{1}{2}) * (((\{b_1,\widehat{y_1}\},\frac{1}{2});(\{e_1,\widehat{z_1}\},\frac{1}{2}))[]((\{y_2\},\frac{1}{2});(\{z_2\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_2\},\frac{1}{2}) * (((\{b_2,\widehat{y_2}\},\frac{1}{2});(\{e_2,\widehat{z_2}\},\frac{1}{2}))[]((\{y_3\},\frac{1}{2});(\{z_3\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_3\},\frac{1}{2}) * (((\{b_3,\widehat{y_3}\},\frac{1}{2});(\{e_3,\widehat{z_3}\},\frac{1}{2}))[]((\{y_4\},\frac{1}{2});(\{z_4\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_4\},\frac{1}{2}) * (((\{b_4,\widehat{y_4}\},\frac{1}{2});(\{e_4,\widehat{z_4}\},\frac{1}{2}))[]((\{y_5\},\frac{1}{2});(\{z_5\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\frac{1}{2}) * (((\{b_5,\widehat{y_5}\},\frac{1}{2});(\{e_5,\widehat{z_5}\},\frac{1}{2}))[]((\{y_1\},\frac{1}{2});(\{z_1\},\frac{1}{2}))) * \mathsf{Stop}]) \\ \mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ x_3\ \mathsf{sy}\ x_4\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ y_3\ \mathsf{sy}\ y_4\ \mathsf{sy}\ y_5\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{sy}\ z_3\ \mathsf{sy}\ z_4\ \mathsf{sy}\ z_5 \\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ x_3\ \mathsf{rs}\ x_4\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ y_3\ \mathsf{rs}\ y_4\ \mathsf{rs}\ y_5\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2\ \mathsf{rs}\ z_3\ \mathsf{rs}\ z_4\ \mathsf{rs}\ z_5]_{\approx}, \end{array}$$

$$\begin{array}{c} s_4 = [([(\{x_1\},\frac{1}{2}) * (((\{b_1,\widehat{y_1}\},\frac{1}{2});(\{e_1,\widehat{z_1}\},\frac{1}{2}))[]((\{y_2\},\frac{1}{2});(\{z_2\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_2\},\frac{1}{2}) * (((\{b_2,\widehat{y_2}\},\frac{1}{2});(\{e_2,\widehat{z_2}\},\frac{1}{2}))[]((\{y_3\},\frac{1}{2});(\{z_3\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_3\},\frac{1}{2}) * (((\{b_3,\widehat{y_3}\},\frac{1}{2});(\{e_3,\widehat{z_3}\},\frac{1}{2}))[]((\{y_4\},\frac{1}{2});(\{z_4\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_4\},\frac{1}{2}) * (((\{b_4,\widehat{y_4}\},\frac{1}{2});(\{e_4,\widehat{z_4}\},\frac{1}{2}))[]((\{y_5\},\frac{1}{2});(\{z_5\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\frac{1}{2}) * (((\{b_5,\widehat{y_5}\},\frac{1}{2});(\{e_5,\widehat{z_5}\},\frac{1}{2}))[]((\{y_1\},\frac{1}{2});(\{z_1\},\frac{1}{2}))) * \mathsf{Stop}]) \\ \mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ x_3\ \mathsf{sy}\ x_4\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ y_3\ \mathsf{sy}\ y_4\ \mathsf{sy}\ y_5\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{sy}\ z_3\ \mathsf{sy}\ z_4\ \mathsf{sy}\ z_5 \\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ x_3\ \mathsf{rs}\ x_4\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ y_3\ \mathsf{rs}\ y_4\ \mathsf{rs}\ y_5\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2\ \mathsf{rs}\ z_3\ \mathsf{rs}\ z_4\ \mathsf{rs}\ z_5]_{\approx}, \end{array}$$

$$\begin{array}{c} s_5 = [([(\{x_1\},\frac{1}{2}) * (((\{b_1,\widehat{y_1}\},\frac{1}{2});(\{e_1,\widehat{z_1}\},\frac{1}{2}))[]((\{y_2\},\frac{1}{2});(\{z_2\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_2\},\frac{1}{2}) * (((\{b_2,\widehat{y_2}\},\frac{1}{2});(\{e_2,\widehat{z_2}\},\frac{1}{2}))[]((\{y_3\},\frac{1}{2});(\{z_3\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_3\},\frac{1}{2}) * (((\{b_3,\widehat{y_3}\},\frac{1}{2});(\{e_3,\widehat{z_3}\},\frac{1}{2}))[]((\{y_4\},\frac{1}{2});(\{z_4\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_4\},\frac{1}{2}) * (((\{b_4,\widehat{y_4}\},\frac{1}{2});(\{e_4,\widehat{z_4}\},\frac{1}{2}))[]((\{y_5\},\frac{1}{2});(\{z_5\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\frac{1}{2}) * (((\{b_5,\widehat{y_5}\},\frac{1}{2});(\{e_5,\widehat{z_5}\},\frac{1}{2}))[]((\{y_1\},\frac{1}{2});(\{z_1\},\frac{1}{2}))) * \mathsf{Stop}]) \\ \mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ x_3\ \mathsf{sy}\ x_4\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ y_3\ \mathsf{sy}\ y_4\ \mathsf{sy}\ y_5\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{sy}\ z_3\ \mathsf{sy}\ z_4\ \mathsf{sy}\ z_5 \\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ x_3\ \mathsf{rs}\ x_4\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ y_3\ \mathsf{rs}\ y_4\ \mathsf{rs}\ y_5\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2\ \mathsf{rs}\ z_3\ \mathsf{rs}\ z_4\ \mathsf{rs}\ z_5]_{\approx}, \end{array}$$

$$\begin{array}{c} s_6 = [([(\{x_1\},\frac{1}{2}) * (((\{b_1,\widehat{y_1}\},\frac{1}{2});(\{e_1,\widehat{z_1}\},\frac{1}{2}))[]((\{y_2\},\frac{1}{2});(\{z_2\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_2\},\frac{1}{2}) * (((\{b_2,\widehat{y_2}\},\frac{1}{2});(\{e_2,\widehat{z_2}\},\frac{1}{2}))[]((\{y_3\},\frac{1}{2});(\{z_3\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_3\},\frac{1}{2}) * (((\{b_3,\widehat{y_3}\},\frac{1}{2});(\{e_3,\widehat{z_3}\},\frac{1}{2}))[]((\{y_4\},\frac{1}{2});(\{z_4\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{x_4\},\frac{1}{2}) * (((\{b_4,\widehat{y_4}\},\frac{1}{2});(\{e_4,\widehat{z_4}\},\frac{1}{2}))[]((\{y_5\},\frac{1}{2});(\{z_5\},\frac{1}{2}))) * \mathsf{Stop}] \| \\ [(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\frac{1}{2}) * (((\{b_5,\widehat{y_5}\},\frac{1}{2});(\{e_5,\widehat{z_5}\},\frac{1}{2}))[]((\{y_1\},\frac{1}{2});(\{z_1\},\frac{1}{2}))) * \mathsf{Stop}]) \\ \mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ x_3\ \mathsf{sy}\ x_4\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ y_3\ \mathsf{sy}\ y_4\ \mathsf{sy}\ y_5\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{sy}\ z_3\ \mathsf{sy}\ z_4\ \mathsf{sy}\ z_5 \\ \mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ x_3\ \mathsf{rs}\ x_4\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ y_3\ \mathsf{rs}\ y_4\ \mathsf{rs}\ y_5\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2\ \mathsf{rs}\ z_3\ \mathsf{rs}\ z_4\ \mathsf{rs}\ z_5]_{\approx}, \end{array}$$

$$s_7 = [([([(\{x_1\},\tfrac{1}{2}) * (((\overline{(\{b_1,\widehat{y_1}\},\tfrac{1}{2})};(\{e_1,\widehat{z_1}\},\tfrac{1}{2}))[]((\{y_2\},\tfrac{1}{2});\underline{(\{z_2\},\tfrac{1}{2})})) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\tfrac{1}{2}) * (((\{b_2,\widehat{y_2}\},\tfrac{1}{2});\underline{(\{e_2,\widehat{z_2}\},\tfrac{1}{2})})[]((\{y_3\},\tfrac{1}{2});(\{z_3\},\tfrac{1}{2}))) * \mathsf{Stop}]\|$$
$$[(\{x_3\},\tfrac{1}{2}) * (((\{b_3,\widehat{y_3}\},\tfrac{1}{2});(\{e_3,\widehat{z_3}\},\tfrac{1}{2}))[](\underline{(\{y_4\},\tfrac{1}{2})};(\{z_4\},\tfrac{1}{2}))) * \mathsf{Stop}]\|$$
$$[(\{x_4\},\tfrac{1}{2}) * (((\{b_4,\widehat{y_4}\},\tfrac{1}{2});(\{e_4,\widehat{z_4}\},\tfrac{1}{2}))[](\overline{(\{y_5\},\tfrac{1}{2})};(\{z_5\},\tfrac{1}{2}))) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\tfrac{1}{2}) * (((\{b_5,\widehat{y_5}\},\tfrac{1}{2});(\{e_5,\widehat{z_5}\},\tfrac{1}{2}))[]((\{y_1\},\tfrac{1}{2});(\{z_1\},\tfrac{1}{2}))) * \mathsf{Stop}])$$
$$\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ x_3\ \mathsf{sy}\ x_4\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ y_3\ \mathsf{sy}\ y_4\ \mathsf{sy}\ y_5\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{sy}\ z_3\ \mathsf{sy}\ z_4\ \mathsf{sy}\ z_5$$
$$\mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ x_3\ \mathsf{rs}\ x_4\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ y_3\ \mathsf{rs}\ y_4\ \mathsf{rs}\ y_5\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2\ \mathsf{rs}\ z_3\ \mathsf{rs}\ z_4\ \mathsf{rs}\ z_5]_{\approx},$$

$$s_8 = [([([(\{x_1\},\tfrac{1}{2}) * (((\{b_1,\widehat{y_1}\},\tfrac{1}{2});\underline{(\{e_1,\widehat{z_1}\},\tfrac{1}{2})})[]((\{y_2\},\tfrac{1}{2});\overline{(\{z_2\},\tfrac{1}{2})})) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\tfrac{1}{2}) * (((\{b_2,\widehat{y_2}\},\tfrac{1}{2});\overline{(\{e_2,\widehat{z_2}\},\tfrac{1}{2})})[]((\{y_3\},\tfrac{1}{2});\underline{(\{z_3\},\tfrac{1}{2})})) * \mathsf{Stop}]\|$$
$$[(\{x_3\},\tfrac{1}{2}) * (((\{b_3,\widehat{y_3}\},\tfrac{1}{2});\underline{(\{e_3,\widehat{z_3}\},\tfrac{1}{2})})[]((\{y_4\},\tfrac{1}{2});\overline{(\{z_4\},\tfrac{1}{2})})) * \mathsf{Stop}]\|$$
$$[(\{x_4\},\tfrac{1}{2}) * (((\{b_4,\widehat{y_4}\},\tfrac{1}{2});\overline{(\{e_4,\widehat{z_4}\},\tfrac{1}{2})})[]((\{y_5\},\tfrac{1}{2});(\{z_5\},\tfrac{1}{2}))) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\tfrac{1}{2}) * (((\{b_5,\widehat{y_5}\},\tfrac{1}{2});(\{e_5,\widehat{z_5}\},\tfrac{1}{2}))[]((\{y_1\},\tfrac{1}{2});(\{z_1\},\tfrac{1}{2}))) * \mathsf{Stop}])$$
$$\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ x_3\ \mathsf{sy}\ x_4\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ y_3\ \mathsf{sy}\ y_4\ \mathsf{sy}\ y_5\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{sy}\ z_3\ \mathsf{sy}\ z_4\ \mathsf{sy}\ z_5$$
$$\mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ x_3\ \mathsf{rs}\ x_4\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ y_3\ \mathsf{rs}\ y_4\ \mathsf{rs}\ y_5\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2\ \mathsf{rs}\ z_3\ \mathsf{rs}\ z_4\ \mathsf{rs}\ z_5]_{\approx},$$

$$s_9 = [([([(\{x_1\},\tfrac{1}{2}) * \overline{(((\{b_1,\widehat{y_1}\},\tfrac{1}{2});(\{e_1,\widehat{z_1}\},\tfrac{1}{2}))[]((\{y_2\},\tfrac{1}{2});(\{z_2\},\tfrac{1}{2})))} * \mathsf{Stop}]\|$$
$$[(\{x_2\},\tfrac{1}{2}) * (((\{b_2,\widehat{y_2}\},\tfrac{1}{2});\underline{(\{e_2,\widehat{z_2}\},\tfrac{1}{2})})[]((\{y_3\},\tfrac{1}{2});\overline{(\{z_3\},\tfrac{1}{2})})) * \mathsf{Stop}]\|$$
$$[(\{x_3\},\tfrac{1}{2}) * (((\{b_3,\widehat{y_3}\},\tfrac{1}{2});\overline{(\{e_3,\widehat{z_3}\},\tfrac{1}{2})})[]((\{y_4\},\tfrac{1}{2});\underline{(\{z_4\},\tfrac{1}{2})})) * \mathsf{Stop}]\|$$
$$[(\{x_4\},\tfrac{1}{2}) * (((\{b_4,\widehat{y_4}\},\tfrac{1}{2});\underline{(\{e_4,\widehat{z_4}\},\tfrac{1}{2})})[]((\{y_5\},\tfrac{1}{2});\overline{(\{z_5\},\tfrac{1}{2})})) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\tfrac{1}{2}) * (((\{b_5,\widehat{y_5}\},\tfrac{1}{2});\overline{(\{e_5,\widehat{z_5}\},\tfrac{1}{2})})[]((\{y_1\},\tfrac{1}{2});(\{z_1\},\tfrac{1}{2}))) * \mathsf{Stop}])$$
$$\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ x_3\ \mathsf{sy}\ x_4\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ y_3\ \mathsf{sy}\ y_4\ \mathsf{sy}\ y_5\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{sy}\ z_3\ \mathsf{sy}\ z_4\ \mathsf{sy}\ z_5$$
$$\mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ x_3\ \mathsf{rs}\ x_4\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ y_3\ \mathsf{rs}\ y_4\ \mathsf{rs}\ y_5\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2\ \mathsf{rs}\ z_3\ \mathsf{rs}\ z_4\ \mathsf{rs}\ z_5]_{\approx},$$

$$s_{10} = [([([(\{x_1\},\tfrac{1}{2}) * (((\{b_1,\widehat{y_1}\},\tfrac{1}{2});(\{e_1,\widehat{z_1}\},\tfrac{1}{2}))[]((\{y_2\},\tfrac{1}{2});\overline{(\{z_2\},\tfrac{1}{2})})) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\tfrac{1}{2}) * (((\{b_2,\widehat{y_2}\},\tfrac{1}{2});\overline{(\{e_2,\widehat{z_2}\},\tfrac{1}{2})})[](\underline{(\{y_3\},\tfrac{1}{2})};(\{z_3\},\tfrac{1}{2}))) * \mathsf{Stop}]\|$$
$$[(\{x_3\},\tfrac{1}{2}) * \underline{(((\{b_3,\widehat{y_3}\},\tfrac{1}{2});(\{e_3,\widehat{z_3}\},\tfrac{1}{2}))[](\overline{(\{y_4\},\tfrac{1}{2})};(\{z_4\},\tfrac{1}{2})))} * \mathsf{Stop}]\|$$
$$[(\{x_4\},\tfrac{1}{2}) * (((\{b_4,\widehat{y_4}\},\tfrac{1}{2});(\{e_4,\widehat{z_4}\},\tfrac{1}{2}))[]((\{y_5\},\tfrac{1}{2});(\{z_5\},\tfrac{1}{2}))) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\tfrac{1}{2}) * ((\overline{(\{b_5,\widehat{y_5}\},\tfrac{1}{2})};(\{e_5,\widehat{z_5}\},\tfrac{1}{2}))[]((\{y_1\},\tfrac{1}{2});(\{z_1\},\tfrac{1}{2}))) * \mathsf{Stop}])$$
$$\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ x_3\ \mathsf{sy}\ x_4\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ y_3\ \mathsf{sy}\ y_4\ \mathsf{sy}\ y_5\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{sy}\ z_3\ \mathsf{sy}\ z_4\ \mathsf{sy}\ z_5$$
$$\mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ x_3\ \mathsf{rs}\ x_4\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ y_3\ \mathsf{rs}\ y_4\ \mathsf{rs}\ y_5\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2\ \mathsf{rs}\ z_3\ \mathsf{rs}\ z_4\ \mathsf{rs}\ z_5]_{\approx},$$

$$s_{11} = [([([(\{x_1\},\tfrac{1}{2}) \underline{* (((\{b_1,\widehat{y_1}\},\tfrac{1}{2});(\{e_1,\widehat{z_1}\},\tfrac{1}{2}))[]}(\overline{(\{y_2\},\tfrac{1}{2})};(\{z_2\},\tfrac{1}{2}))) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\tfrac{1}{2}) * ((\underline{(\{b_2,\widehat{y_2}\},\tfrac{1}{2})};(\{e_2,\widehat{z_2}\},\tfrac{1}{2}))[]((\{y_3\},\tfrac{1}{2});(\{z_3\},\tfrac{1}{2}))) * \mathsf{Stop}]\|$$
$$[(\{x_3\},\tfrac{1}{2}) * (((\{b_3,\widehat{y_3}\},\tfrac{1}{2});(\{e_3,\widehat{z_3}\},\tfrac{1}{2}))[]((\{y_4\},\tfrac{1}{2});\underline{(\{z_4\},\tfrac{1}{2})})) * \mathsf{Stop}]\|$$
$$[(\{x_4\},\tfrac{1}{2}) * (((\{b_4,\widehat{y_4}\},\tfrac{1}{2});\underline{(\{e_4,\widehat{z_4}\},\tfrac{1}{2})})[]((\{y_5\},\tfrac{1}{2});\overline{(\{z_5\},\tfrac{1}{2})})) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\tfrac{1}{2}) * (((\{b_5,\widehat{y_5}\},\tfrac{1}{2});\overline{(\{e_5,\widehat{z_5}\},\tfrac{1}{2})})[]((\{y_1\},\tfrac{1}{2});(\{z_1\},\tfrac{1}{2}))) * \mathsf{Stop}])$$
$$\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ x_3\ \mathsf{sy}\ x_4\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ y_3\ \mathsf{sy}\ y_4\ \mathsf{sy}\ y_5\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{sy}\ z_3\ \mathsf{sy}\ z_4\ \mathsf{sy}\ z_5$$
$$\mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ x_3\ \mathsf{rs}\ x_4\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ y_3\ \mathsf{rs}\ y_4\ \mathsf{rs}\ y_5\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2\ \mathsf{rs}\ z_3\ \mathsf{rs}\ z_4\ \mathsf{rs}\ z_5]_{\approx},$$

$$s_{12} = [([([(\{x_1\},\tfrac{1}{2}) * (((\{b_1,\widehat{y_1}\},\tfrac{1}{2});\underline{(\{e_1,\widehat{z_1}\},\tfrac{1}{2})})[]((\{y_2\},\tfrac{1}{2});\overline{(\{z_2\},\tfrac{1}{2})})) * \mathsf{Stop}]\|$$
$$[(\{x_2\},\tfrac{1}{2}) * \underline{(((\{b_2,\widehat{y_2}\},\tfrac{1}{2});\overline{(\{e_2,\widehat{z_2}\},\tfrac{1}{2})})[]((\{y_3\},\tfrac{1}{2});(\{z_3\},\tfrac{1}{2})))} * \mathsf{Stop}]\|$$
$$[(\{x_3\},\tfrac{1}{2}) * (((\{b_3,\widehat{y_3}\},\tfrac{1}{2});(\{e_3,\widehat{z_3}\},\tfrac{1}{2}))[]((\{y_4\},\tfrac{1}{2});\underline{(\{z_4\},\tfrac{1}{2})})) * \mathsf{Stop}]\|$$
$$[(\{x_4\},\tfrac{1}{2}) * (((\{b_4,\widehat{y_4}\},\tfrac{1}{2});\underline{(\{e_4,\widehat{z_4}\},\tfrac{1}{2})})[]((\{y_5\},\tfrac{1}{2});\overline{(\{z_5\},\tfrac{1}{2})})) * \mathsf{Stop}]\|$$
$$[(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\tfrac{1}{2}) * (((\{b_5,\widehat{y_5}\},\tfrac{1}{2});\overline{(\{e_5,\widehat{z_5}\},\tfrac{1}{2})})[]((\{y_1\},\tfrac{1}{2});(\{z_1\},\tfrac{1}{2}))) * \mathsf{Stop}])$$
$$\mathsf{sy}\ x_1\ \mathsf{sy}\ x_2\ \mathsf{sy}\ x_3\ \mathsf{sy}\ x_4\ \mathsf{sy}\ y_1\ \mathsf{sy}\ y_2\ \mathsf{sy}\ y_3\ \mathsf{sy}\ y_4\ \mathsf{sy}\ y_5\ \mathsf{sy}\ z_1\ \mathsf{sy}\ z_2\ \mathsf{sy}\ z_3\ \mathsf{sy}\ z_4\ \mathsf{sy}\ z_5$$
$$\mathsf{rs}\ x_1\ \mathsf{rs}\ x_2\ \mathsf{rs}\ x_3\ \mathsf{rs}\ x_4\ \mathsf{rs}\ y_1\ \mathsf{rs}\ y_2\ \mathsf{rs}\ y_3\ \mathsf{rs}\ y_4\ \mathsf{rs}\ y_5\ \mathsf{rs}\ z_1\ \mathsf{rs}\ z_2\ \mathsf{rs}\ z_3\ \mathsf{rs}\ z_4\ \mathsf{rs}\ z_5]_{\approx}.$$

The states are interpreted as follows: s_1 is the initial state, s_2: the system is activated and no philosophers dine, s_3: philosopher 1 dines, s_4: philosophers 1 and 4 dine, s_5: philosophers 1 and 3 dine, s_6: philosopher 4 dines, s_7: philosopher 3 dines, s_8: philosophers 2 and 4 dine, s_9: philosophers 3 and 5 dine, s_{10}: philosopher 2 dines, s_{11}: philosopher 5 dines, s_{12}: philosophers 2 and 5 dine.

In Figure 35, the transition system without empty loops $TS^*(\overline{E})$ is presented. In Figure 36, the underlying DTMC without empty loops $DTMC^*(\overline{E})$ is depicted.

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

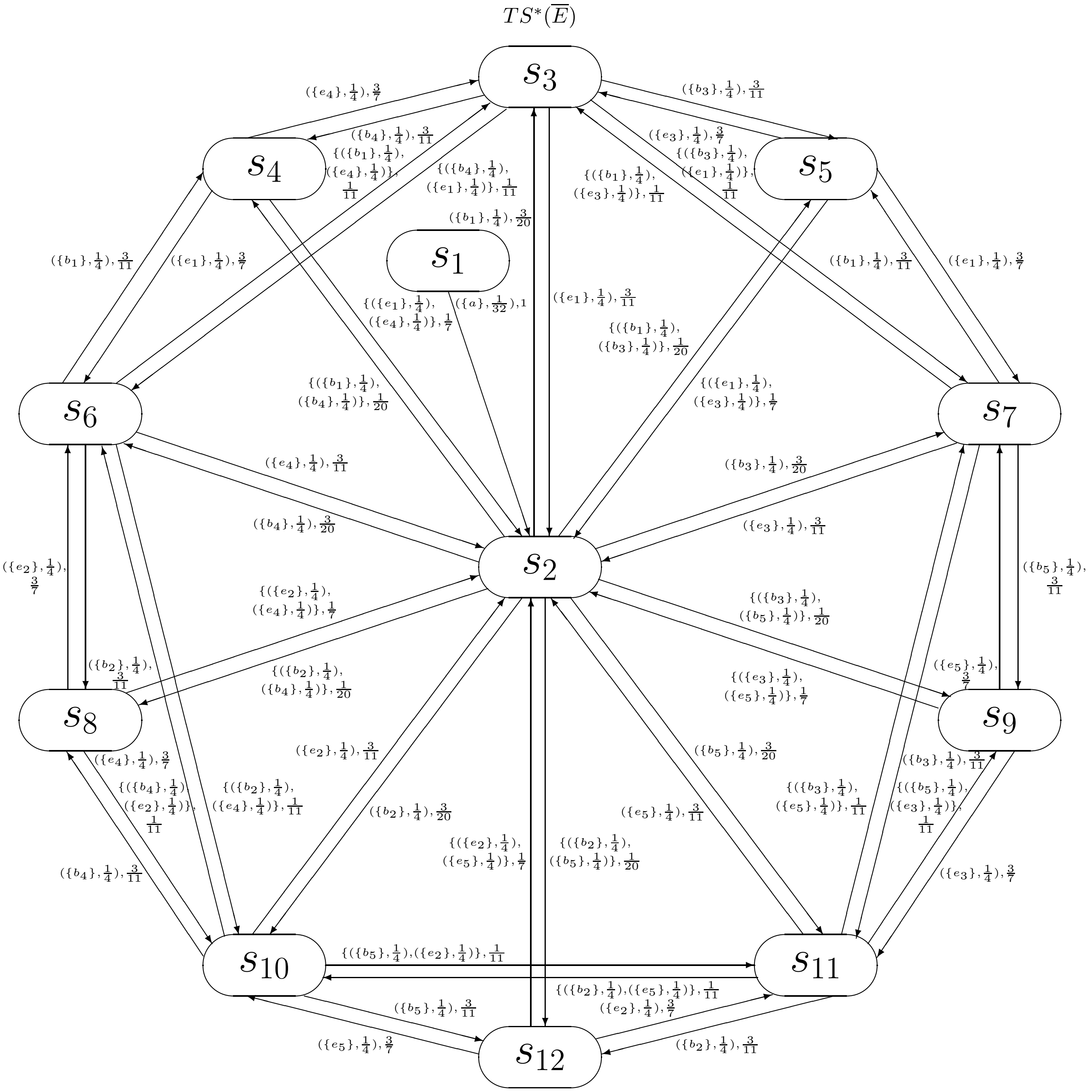

Figure 35: The transition system without empty loops of the dining philosophers system

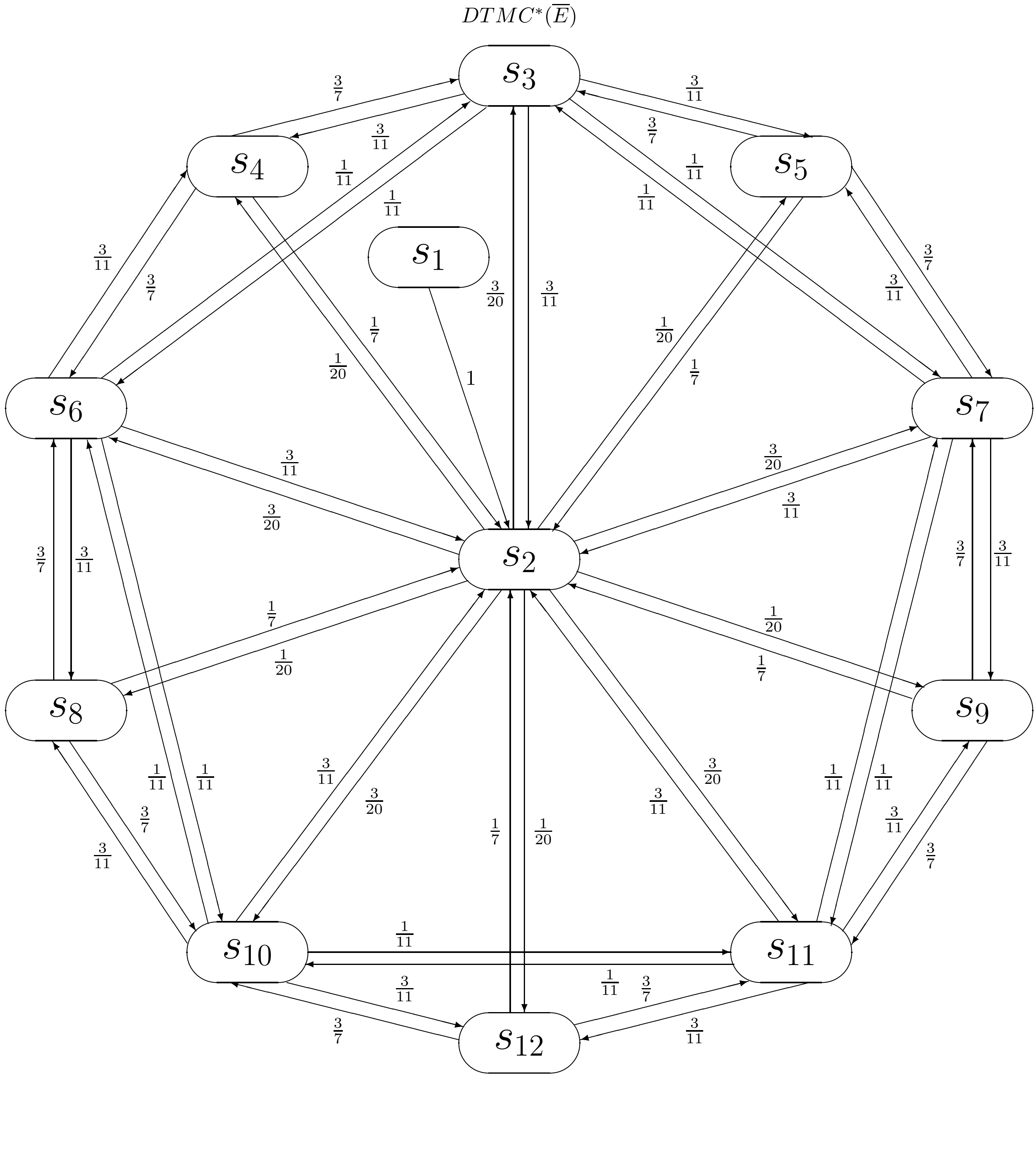

Figure 36: The underlying DTMC without empty loops of the dining philosophers system

Table 10: Transient and steady-state probabilities of the dining philosophers system

k	0	1	2	3	4	5	6	7	8	9	10	∞
$\psi_1^*[k]$	1	0	0	0	0	0	0	0	0	0	0	0
$\psi_2^*[k]$	0	1	0	0.2403	0.1541	0.1981	0.1716	0.1884	0.1776	0.1846	0.1800	0.1818
$\psi_3^*[k]$	0	0	0.1500	0.0701	0.1189	0.0878	0.1079	0.0949	0.1033	0.0979	0.1014	0.1000
$\psi_4^*[k]$	0	0	0.0500	0.0818	0.0503	0.0726	0.0578	0.0674	0.0612	0.0652	0.0626	0.0636

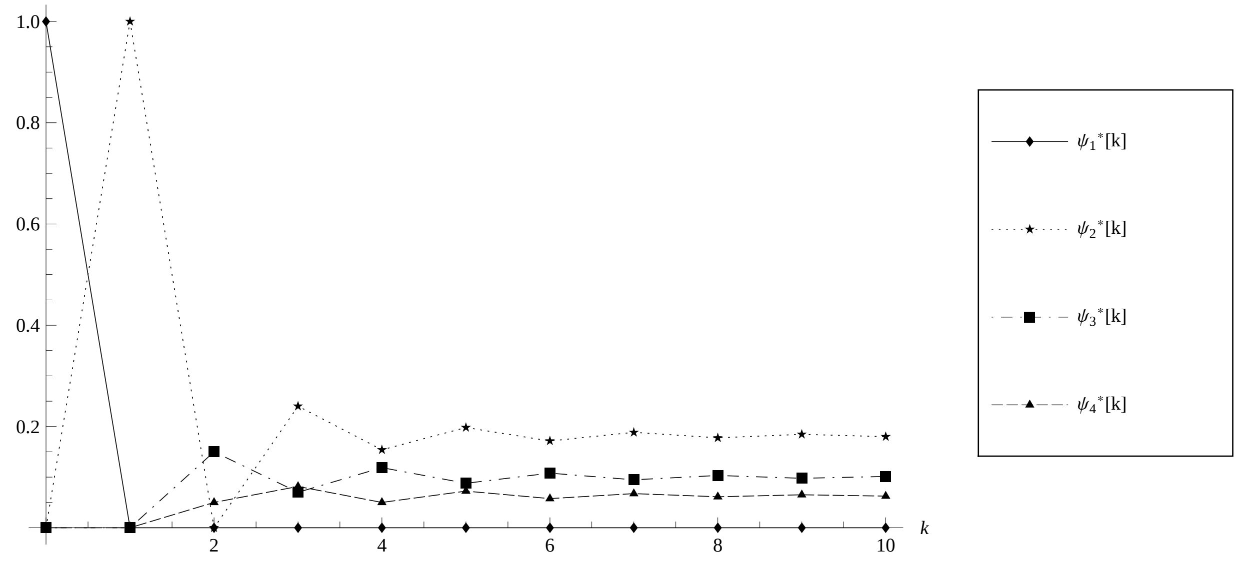

Figure 37: Transient probabilities alteration diagram of the dining philosophers system

$$\mathbf{P}^* = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{3}{20} & \frac{1}{20} & \frac{1}{20} & \frac{3}{20} & \frac{3}{20} & \frac{1}{20} & \frac{1}{20} & \frac{3}{20} & \frac{3}{20} & \frac{1}{20} \\ 0 & \frac{3}{11} & 0 & \frac{3}{11} & \frac{3}{11} & \frac{1}{11} & \frac{1}{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{7} & \frac{3}{7} & 0 & 0 & \frac{3}{7} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{7} & \frac{3}{7} & 0 & 0 & 0 & \frac{3}{7} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{3}{11} & \frac{1}{11} & \frac{3}{11} & 0 & 0 & 0 & \frac{3}{11} & 0 & \frac{1}{11} & 0 & 0 \\ 0 & \frac{3}{11} & \frac{1}{11} & 0 & \frac{3}{11} & 0 & 0 & 0 & \frac{3}{11} & 0 & \frac{1}{11} & 0 \\ 0 & \frac{1}{7} & 0 & 0 & 0 & \frac{3}{7} & 0 & 0 & 0 & \frac{3}{7} & 0 & 0 \\ 0 & \frac{1}{7} & 0 & 0 & 0 & 0 & \frac{3}{7} & 0 & 0 & 0 & \frac{3}{7} & 0 \\ 0 & \frac{3}{11} & 0 & 0 & 0 & \frac{1}{11} & 0 & \frac{3}{11} & 0 & 0 & \frac{1}{11} & \frac{3}{11} \\ 0 & \frac{3}{11} & 0 & 0 & 0 & 0 & \frac{1}{11} & 0 & \frac{3}{11} & \frac{1}{11} & 0 & \frac{3}{11} \\ 0 & \frac{1}{7} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{7} & \frac{3}{7} & 0 \end{pmatrix}.$$

In Table 10, the transient and the steady-state probabilities $\psi_i^*[k]$ $(1 \leq i \leq 4)$ of the dining philosophers system at the time moments k $(0 \leq k \leq 10)$ and $k = \infty$ are presented, and in Figure 37, the alteration diagram (evolution in time) for the transient probabilities is depicted. It is sufficient to consider the probabilities for the states $s_1, \ldots, s_4$ only, since the corresponding values coincide for $s_3, s_6, s_7, s_{10}, s_{11}$, as well as for $s_4, s_5, s_8, s_9, s_{12}$.

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

$$\psi^* = \left(0, \frac{2}{11}, \frac{1}{10}, \frac{7}{110}, \frac{7}{110}, \frac{1}{10}, \frac{1}{10}, \frac{7}{110}, \frac{7}{110}, \frac{1}{10}, \frac{1}{10}, \frac{7}{110}\right).$$

Note that we do not have the problem of *individual starvation* in our model, since, for each philosopher, the time intervals when he does not eat are finite. In other words, there are no infinite time periods when the philosopher does not eat. It means that his steady-state probability to eat is not equal to zero. For example, the first philosopher eats in the states s_3, s_4, s_5, hence, the stationary probability that he eats is

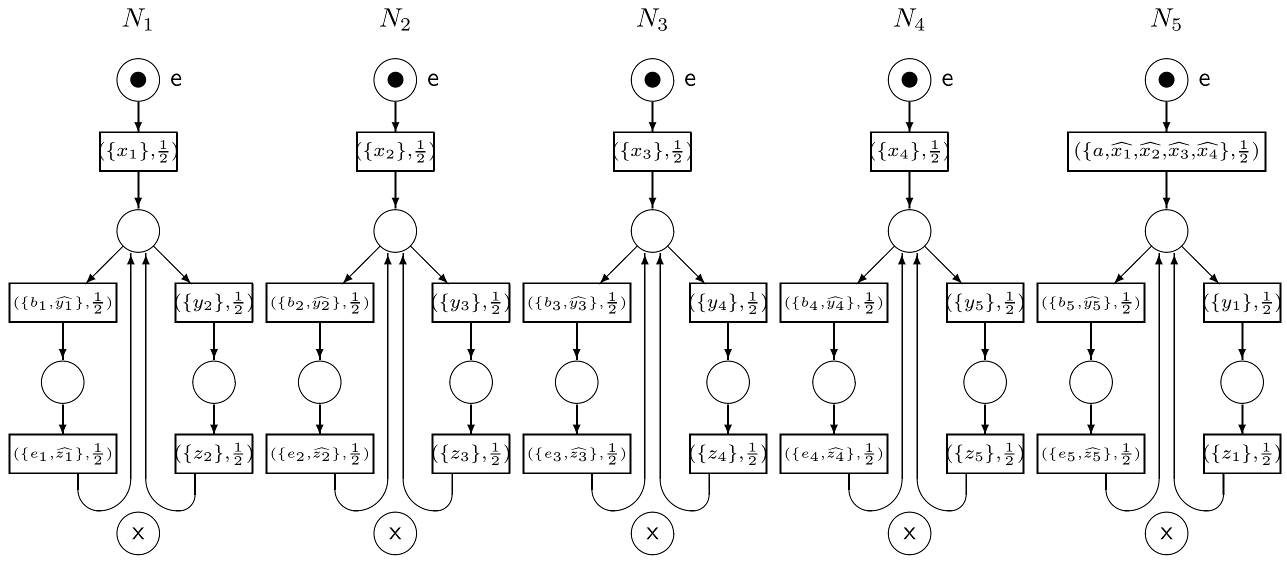

Figure 38: The marked dts-boxes of the dining philosophers

$\psi_3^*+\psi_4^*+\psi_5^*=\frac{1}{10}+\frac{7}{110}+\frac{7}{110}=\frac{5}{21}$. Thus, the first philosopher eats with a positive probability in the long-time behaviour of the system. The argumentation for other philosophers is the same.

We can now calculate the main performance indices.

- The average recurrence time in the state s_2, where all the forks are available, called the *average system run-through*, is $\frac{1}{\psi_2^*}=\frac{11}{2}=5\frac{1}{2}$.
- Nobody eats in the state s_2. Then, the *fraction of time when no philosophers dine* is $\psi_2^*=\frac{2}{11}$.

  Only one philosopher eats in the states $s_3,s_6,s_7,s_{10},s_{11}$. Then, the *fraction of time when only one philosopher dines* is $\psi_3^*+\psi_6^*+\psi_7^*+\psi_{10}^*+\psi_{11}^*=\frac{1}{10}+\frac{1}{10}+\frac{1}{10}+\frac{1}{10}+\frac{1}{10}=\frac{1}{2}$.

  Two philosophers eat together in the states s_4,s_5,s_8,s_9,s_{12}. Then, the *fraction of time when two philosophers dine* is $\psi_4^*+\psi_5^*+\psi_8^*+\psi_9^*+\psi_{12}^*=\frac{7}{110}+\frac{7}{110}+\frac{7}{110}+\frac{7}{110}+\frac{7}{110}=\frac{7}{22}$.

  The *relative fraction of time when two philosophers dine with respect to when only one philosopher dines* is $\frac{7}{22}\cdot\frac{2}{1}=\frac{7}{11}$.
- The beginning of eating of first philosopher $(\{b_1\},\frac{1}{4})$ is only possible from the states s_2,s_6,s_7. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of activities containing $(\{b_1\},\frac{1}{4})$. Thus, the *steady-state probability of the beginning of eating of first philosopher* is $\psi_2^*\sum_{\{\Gamma\mid(\{b_1\},\frac{1}{4})\in\Gamma\}}PT^*(\Gamma,s_2)+\psi_6^*\sum_{\{\Gamma\mid(\{b_1\},\frac{1}{4})\in\Gamma\}}PT^*(\Gamma,s_6)+\psi_7^*\sum_{\{\Gamma\mid(\{b_1\},\frac{1}{4})\in\Gamma\}}PT^*(\Gamma,s_7)=\frac{2}{11}\left(\frac{3}{20}+\frac{1}{20}+\frac{1}{20}\right)+\frac{1}{10}\left(\frac{3}{11}+\frac{1}{11}\right)+\frac{1}{10}\left(\frac{3}{11}+\frac{1}{11}\right)=\frac{13}{110}$.

In Figure 38, the marked dts-boxes corresponding to the dynamic expressions of the dining philosophers are presented, i.e. $N_i=Box_{dts}(\overline{E_i})$ $(1\le i\le 5)$. In Figure 39, the marked dts-box corresponding to the dynamic expression of the dining philosophers system is depicted, i.e. $N=Box_{dts}(\overline{E})$.

10.2.2 The abstract system and its reductions

Let us consider a modification of the dining philosophers system with abstraction from personalities, i.e. such that all the philosophers are indistinguishable. For example, we can just see that one or two philosophers dine but cannot observe who they are. We call this system the abstract dining philosophers one. To implement the abstraction, we replace the actions b_i and e_i $(1\le i\le 5)$ in the system specification by b and e, respectively.

The static expression of the philosopher i $(1\le i\le 4)$ is $F_i=[(\{x_i\},\frac{1}{2})*(((\{b,\widehat{y_i}\},\frac{1}{2});(\{e,\widehat{z_i}\},\frac{1}{2}))[]((\{y_{i+1}\},\frac{1}{2});(\{z_{i+1}\},\frac{1}{2})))*\mathsf{Stop}]$. The static expression of the philosopher 5 is $F_5=[(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\frac{1}{2})*(((\{b,\widehat{y_5}\},\frac{1}{2});(\{e,\widehat{z_5}\},\frac{1}{2}))[]((\{y_1\},\frac{1}{2});(\{z_1\},\frac{1}{2})))*\mathsf{Stop}]$. The static expression of the abstract dining philosophers system is $F=(F_1\|F_2\|F_3\|F_4\|F_5)$ sy x_1 sy x_2 sy x_3 sy x_4 sy y_1 sy y_2 sy y_3 sy y_4 sy y_5 sy z_1 sy z_2 sy z_3 sy z_4 sy z_5 rs x_1 rs x_2 rs x_3 rs x_4 rs y_1 rs y_2 rs y_3 rs y_4 rs y_5 rs z_1 rs z_2 rs z_3 rs z_4 rs z_5.

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

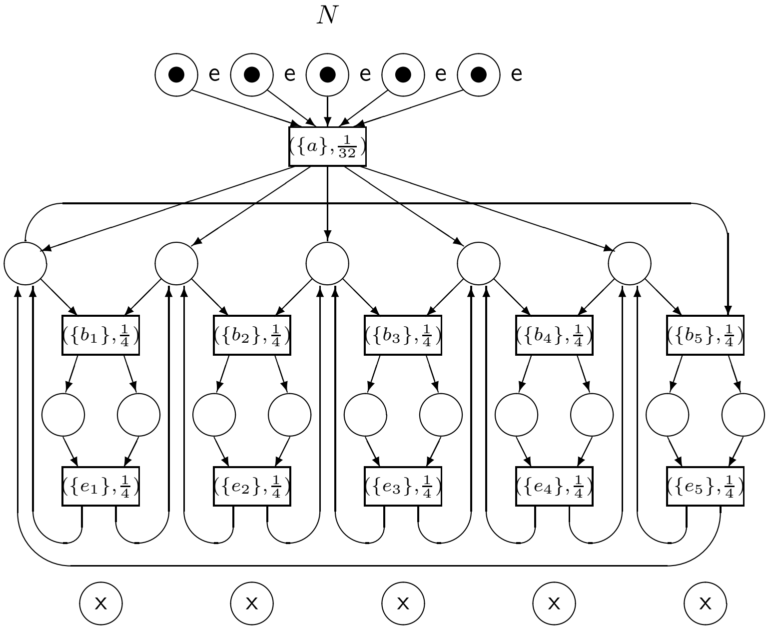

Figure 39: The marked dts-box of the dining philosophers system

The first performance index and the second group of the indices are the same for the standard and the abstract systems. Let us consider the following performance index based on non-personalized viewpoint to the philosophers.

- The beginning of eating of a philosopher $(\{b\},\frac{1}{4})$ is only possible from the states $s_2, s_3, s_6, s_7, s_{10}, s_{11}$. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of activities containing $(\{b\},\frac{1}{4})$. Thus, the *steady-state probability of the beginning of eating of a philosopher* is
$\psi_2^*\sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}}PT^*(\Gamma,s_2)+\psi_3^*\sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}}PT^*(\Gamma,s_3)+\psi_6^*\sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}}PT^*(\Gamma,s_6)+$
$\psi_7^*\sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}}PT^*(\Gamma,s_7)+\psi_{10}^*\sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}}PT^*(\Gamma,s_{10})+\psi_{11}^*\sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}}PT^*(\Gamma,s_{11})=$
$\frac{2}{11}\left(\frac{3}{20}+\frac{1}{20}+\frac{3}{20}+\frac{1}{20}+\frac{3}{20}+\frac{1}{20}+\frac{3}{20}+\frac{1}{20}+\frac{3}{20}+\frac{1}{20}\right)+\frac{1}{4}\left(\frac{3}{11}+\frac{1}{11}+\frac{3}{11}+\frac{1}{11}\right)+\frac{1}{4}\left(\frac{3}{11}+\frac{1}{11}+\frac{3}{11}+\frac{1}{11}\right)+$
$\frac{1}{4}\left(\frac{3}{11}+\frac{1}{11}+\frac{3}{11}+\frac{1}{11}\right)+\frac{1}{4}\left(\frac{3}{11}+\frac{1}{11}+\frac{3}{11}+\frac{1}{11}\right)+\frac{1}{4}\left(\frac{3}{11}+\frac{1}{11}+\frac{3}{11}+\frac{1}{11}\right)=\frac{6}{11}.$

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

Let us consider a reduction of the abstract dining philosophers system. The static expression of the philosopher 1 is $F_1' = [(\{x\},\frac{1}{2}) * ((\{b\},\frac{2}{5});(\{e\},\frac{1}{4})) * \mathsf{Stop}]$. The static expression of the philosopher 2 is $F_2' = [(\{a,\hat{x}\},\frac{1}{16}) * ((\{b\},\frac{2}{5});(\{e\},\frac{1}{4})) * \mathsf{Stop}]$. The static expression of the reduced abstract dining philosophers system is $F' = (F_1' \| F_2')$ sy x rs x.

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

$$s_1' = [([\overline{(\{x\},\tfrac{1}{2})} * ((\{b\},\tfrac{2}{5})_1;(\{e\},\tfrac{1}{4})_1) * \mathsf{Stop}] \| [\overline{(\{a,\hat{x}\},\tfrac{1}{16})} * ((\{b\},\tfrac{2}{5})_2;(\{e\},\tfrac{1}{4})_2) * \mathsf{Stop}]) \text{ sy } x \text{ rs } x]_{\approx},$$
$$s_2' = [([(\{x\},\tfrac{1}{2}) * (\overline{(\{b\},\tfrac{2}{5})_1};(\{e\},\tfrac{1}{4})_1) * \mathsf{Stop}] \| [(\{a,\hat{x}\},\tfrac{1}{16}) * (\overline{(\{b\},\tfrac{2}{5})_2};(\{e\},\tfrac{1}{4})_2) * \mathsf{Stop}]) \text{ sy } x \text{ rs } x]_{\approx},$$
$$s_3' = [([(\{x\},\tfrac{1}{2}) * ((\{b\},\tfrac{2}{5})_1;\overline{(\{e\},\tfrac{1}{4})_1}) * \mathsf{Stop}] \| [(\{a,\hat{x}\},\tfrac{1}{16}) * (\overline{(\{b\},\tfrac{2}{5})_2};(\{e\},\tfrac{1}{4})_2) * \mathsf{Stop}]) \text{ sy } x \text{ rs } x]_{\approx},$$
$$s_4' = [([(\{x\},\tfrac{1}{2}) * (\overline{(\{b\},\tfrac{2}{5})_1};(\{e\},\tfrac{1}{4})_1) * \mathsf{Stop}] \| [(\{a,\hat{x}\},\tfrac{1}{16}) * ((\{b\},\tfrac{2}{5})_2;\overline{(\{e\},\tfrac{1}{4})_2}) * \mathsf{Stop}]) \text{ sy } x \text{ rs } x]_{\approx},$$
$$s_5' = [([(\{x\},\tfrac{1}{2}) * ((\{b\},\tfrac{2}{5})_1;\overline{(\{e\},\tfrac{1}{4})_1}) * \mathsf{Stop}] \| [(\{a,\hat{x}\},\tfrac{1}{16}) * ((\{b\},\tfrac{2}{5})_2;\overline{(\{e\},\tfrac{1}{4})_2}) * \mathsf{Stop}]) \text{ sy } x \text{ rs } x]_{\approx}.$$

The states are interpreted as follows: s_1' is the initial state, s_2': the system is activated and no philosophers dine, s_3', s_4': one philosopher dines, s_5': two philosophers dine.

Consider the equivalence $\mathcal{R} : \overline{F} \underline{\leftrightarrow}_{ss} \overline{F'}$ such that $(DR(\overline{F}) \cup DR(\overline{F'}))/_{\mathcal{R}} = \{\mathcal{H}_1,\mathcal{H}_2,\mathcal{H}_3,\mathcal{H}_4\}$, where $\mathcal{H}_1 = \{s_1, s_1'\}$ (the initial state), $\mathcal{H}_2 = \{s_2, s_2'\}$ (the system is activated and no philosophers dine), $\mathcal{H}_3 = \{s_3, s_6, s_7, s_{10}, s_{11}, s_3', s_4'\}$ (one philosopher dines), $\mathcal{H}_4 = \{s_4, s_5, s_8, s_9, s_{12}, s_5'\}$ (two philosophers dine). One can see that F' is a reduction of F with respect to $\underline{\leftrightarrow}_{ss}$.

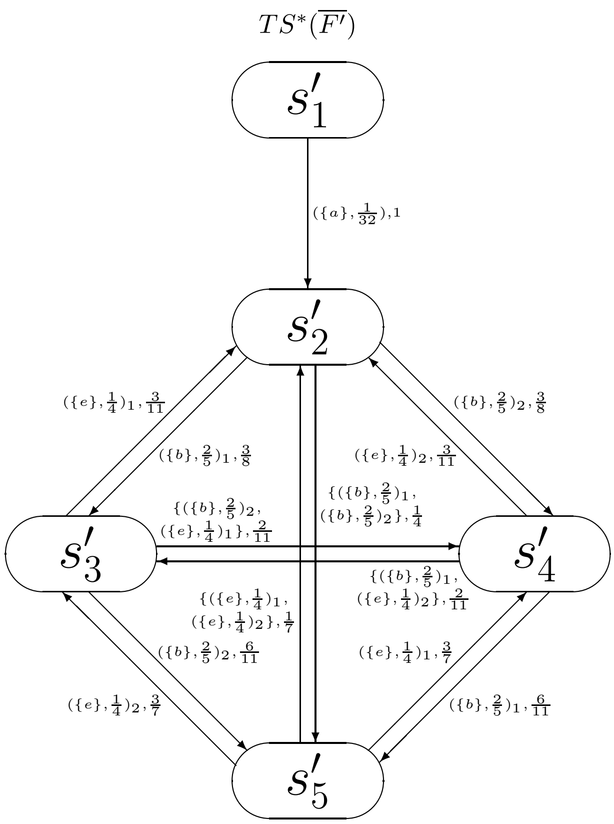

Figure 40: The transition system without empty loops of the reduced abstract dining philosophers system

Table 11: Transient and steady-state probabilities of the reduced abstract dining philosophers system

k	0	1	2	3	4	5	6	7	8	9	10	∞
${\psi'_1}^*[k]$	1	0	0	0	0	0	0	0	0	0	0	0
${\psi'_2}^*[k]$	0	1	0	0.2403	0.1541	0.1981	0.1716	0.1884	0.1776	0.1846	0.1800	0.1818
${\psi'_3}^*[k]$	0	0	0.3750	0.1753	0.2973	0.2195	0.2697	0.2372	0.2583	0.2446	0.2535	0.2500
${\psi'_5}^*[k]$	0	0	0.2500	0.4091	0.2513	0.3628	0.2890	0.3371	0.3059	0.3261	0.3130	0.3182

In Figure 40, the transition system without empty loops $TS^*(\overline{F'})$ is presented. In Figure 41, the underlying DTMC without empty loops $DTMC^*(\overline{F'})$ is depicted.

The TPM for $DTMC^*(\overline{F'})$ is

$$\mathbf{P}'^* = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{3}{8} & \frac{3}{8} & \frac{1}{4} \\ 0 & \frac{3}{11} & 0 & \frac{2}{11} & \frac{6}{11} \\ 0 & \frac{3}{11} & \frac{2}{11} & 0 & \frac{6}{11} \\ 0 & \frac{1}{7} & \frac{3}{7} & \frac{3}{7} & 0 \end{pmatrix}.$$

In Table 11, the transient and the steady-state probabilities ${\psi'_i}^*[k]$ $(i \in \{1,2,3,5\})$ of the reduced abstract dining philosophers system at the time moments k $(0 \le k \le 10)$ and $k = \infty$ are presented, and in Figure 42, the alteration diagram (evolution in time) for the transient probabilities is depicted. It is sufficient to consider the probabilities for the states s'_1, s'_2, s'_3, s'_5 only, since the corresponding values coincide for s'_3, s'_4.

The steady-state PMF for $DTMC^*(\overline{F'})$ is

$$\psi'^* = \left(0, \frac{2}{11}, \frac{1}{4}, \frac{1}{4}, \frac{7}{22}\right).$$

We can now calculate the main performance indices.

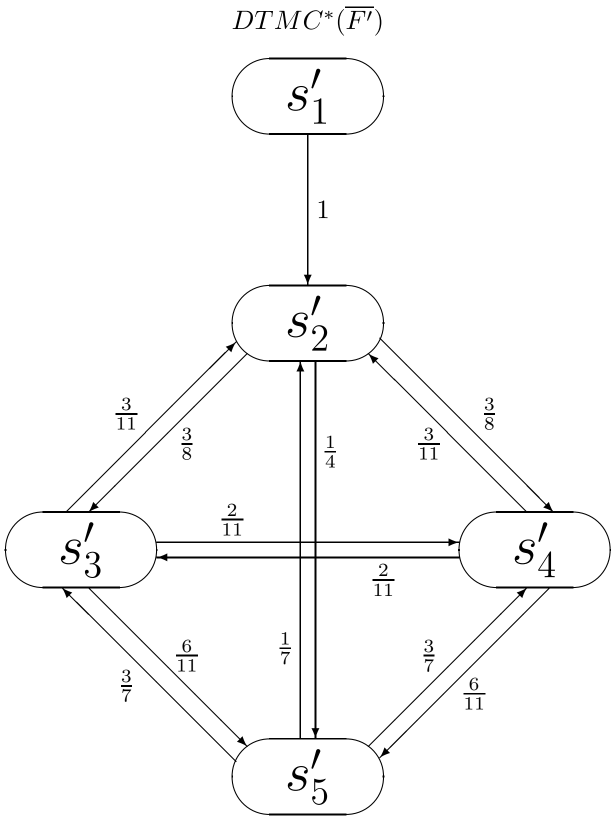

Figure 41: The underlying DTMC without empty loops of the reduced abstract dining philosophers system

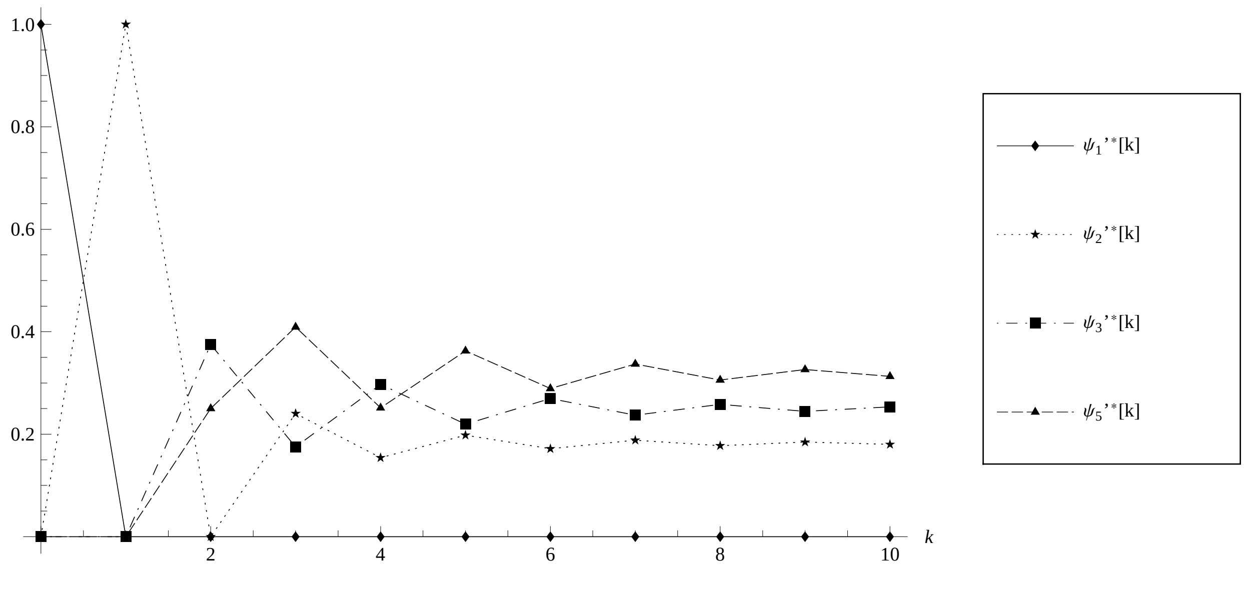

Figure 42: Transient probabilities alteration diagram of the reduced abstract dining philosophers system

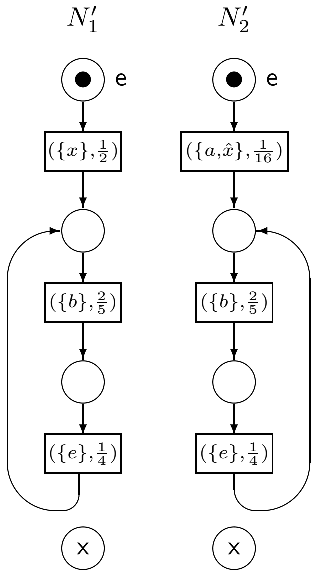

Figure 43: The marked dts-boxes of the reduced abstract dining philosophers

- The average recurrence time in the state s'_2, where all the forks are available, called the *average system run-through*, is $\frac{1}{{\psi'_2}^*} = \frac{11}{2} = 5\frac{1}{2}$.
- Nobody eats in the state s'_2. Then, the *fraction of time when no philosophers dine* is ${\psi'_2}^* = \frac{2}{11}$.

  Only one philosopher eats in the states s'_3, s'_4. Then, the *fraction of time when only one philosopher dines* is ${\psi'_3}^* + {\psi'_4}^* = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$.

  Two philosophers eat together in the state s'_5. Then, the *fraction of time when two philosophers dine* is ${\psi'_5}^* = \frac{7}{22}$.

  The *relative fraction of time when two philosophers dine with respect to when only one philosopher dines* is $\frac{7}{22} \cdot \frac{2}{1} = \frac{7}{11}$.
- The beginning of eating of a philosopher $(\{b\}, \frac{2}{5})$ is only possible from the states s'_2, s'_3, s'_4. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of activities containing $(\{b\}, \frac{2}{5})$. Thus, the *steady-state probability of the beginning of eating of a philosopher* is ${\psi'_2}^* \sum_{\{\Gamma|(\{b\},\frac{2}{5})\in\Gamma\}} PT^*(\Gamma, s'_2) + {\psi'_3}^* \sum_{\{\Gamma|(\{b\},\frac{2}{5})\in\Gamma\}} PT^*(\Gamma, s'_3) + {\psi'_4}^* \sum_{\{\Gamma|(\{b\},\frac{2}{5})\in\Gamma\}} PT^*(\Gamma, s'_4) = \frac{2}{11}\left(\frac{3}{8} + \frac{3}{8} + \frac{1}{4}\right) + \frac{1}{4}\left(\frac{6}{11} + \frac{2}{11}\right) + \frac{1}{4}\left(\frac{6}{11} + \frac{2}{11}\right) = \frac{6}{11}$.

One can see that the performance indices are the same for the complete and the reduced abstract dining philosophers systems. The coincidence of the first performance index, as well as the second group of indices obviously illustrates the result of Proposition 8.1. The coincidence of the third performance index is due to Theorem 8.2: one should just apply its result to the step traces $\{\{b\}\}$, $\{\{b\},\{b\}\}$, $\{\{b\},\{e\}\}$ of the expressions $\overline{F}$ and $\overline{F'}$, and then sum the left and right parts of the three resulting equalities.

In Figure 43, the marked dts-boxes corresponding to the dynamic expressions of the reduced abstract dining philosophers are presented, i.e. $N'_i = Box_{dts}(\overline{F'_i})$ $(1 \leq i \leq 2)$. In Figure 44, the marked dts-box corresponding to the dynamic expression of the reduced abstract dining philosophers system is depicted, i.e. $N' = Box_{dts}(\overline{F'})$.

Note that $TS^*(\overline{F'})$ can be reduced further by merging the equivalent states s'_3 and s'_4, thus, it can be transformed into a transition system with four states only. But the resulting reduction of the initial transition system $TS^*(\overline{F})$ will not correspond to some *dtsPBC* expression anymore.

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

In Figure 45, the quotient transition system without empty loops $TS^*_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is presented. In Figure 46, the quotient underlying DTMC without empty loops $DTMC^*_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is depicted.

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

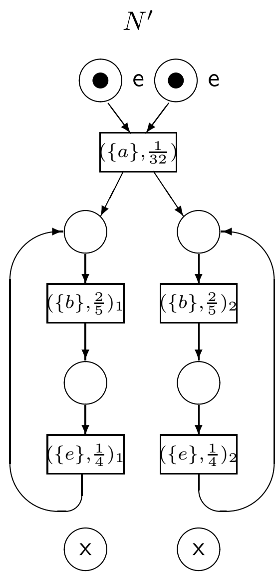

Figure 44: The marked dts-box of the reduced abstract dining philosophers system

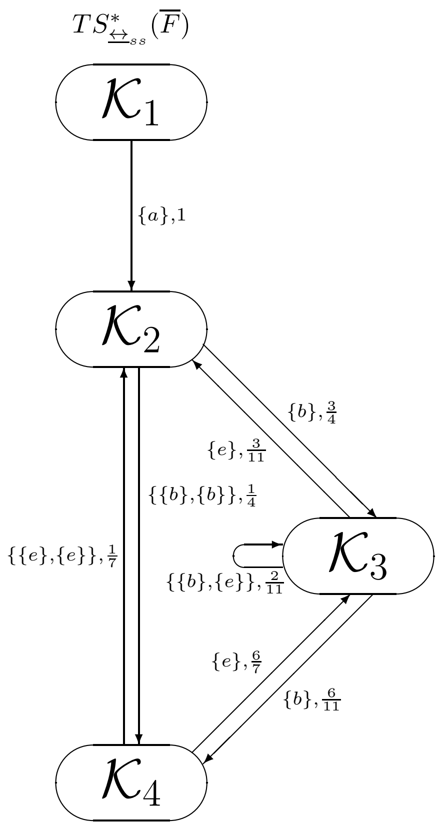

Figure 45: The quotient transition system without empty loops of the abstract dining philosophers system

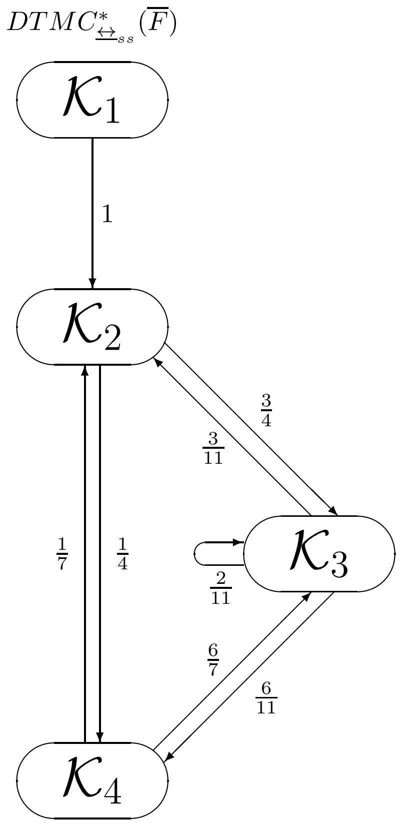

Figure 46: The quotient underlying DTMC without empty loops of the abstract dining philosophers system

Table 12: Transient and steady-state probabilities of the quotient abstract dining philosophers system

k	0	1	2	3	4	5	6	7	8	9	10	∞
$\psi_1''^*[k]$	1	0	0	0	0	0	0	0	0	0	0	0
$\psi_2''^*[k]$	0	1	0	0.2403	0.1541	0.1981	0.1716	0.1884	0.1776	0.1846	0.1800	0.1818
$\psi_3''^*[k]$	0	0	0.7500	0.3506	0.5946	0.4391	0.5394	0.4745	0.5165	0.4893	0.5069	0.5000
$\psi_4''^*[k]$	0	0	0.2500	0.4091	0.2513	0.3628	0.2890	0.3371	0.3059	0.3261	0.3130	0.3182

$$\mathbf{P}''^* = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{3}{4} & \frac{1}{4} \\ 0 & \frac{3}{11} & \frac{2}{11} & \frac{6}{11} \\ 0 & \frac{1}{7} & \frac{6}{7} & 0 \end{pmatrix}.$$

In Table 12, the transient and the steady-state probabilities $\psi_i''^*[k]$ $(1 \leq i \leq 4)$ of the quotient abstract dining philosophers system at the time moments k $(0 \leq k \leq 10)$ and $k = \infty$ are presented, and in Figure 47, the alteration diagram (evolution in time) for the transient probabilities is depicted.

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

$$\psi''^* = \left(0, \frac{2}{11}, \frac{1}{2}, \frac{7}{22}\right).$$

We can now calculate the main performance indices.

- The average recurrence time in the state $\mathcal{K}_2$, where all the forks are available, called the *average system run-through*, is $\frac{1}{\psi_2''^*} = \frac{11}{2} = 5\frac{1}{2}$.

- Nobody eats in the state $\mathcal{K}_2$. Then, the *fraction of time when no philosophers dine* is $\psi_2''^* = \frac{2}{11}$.

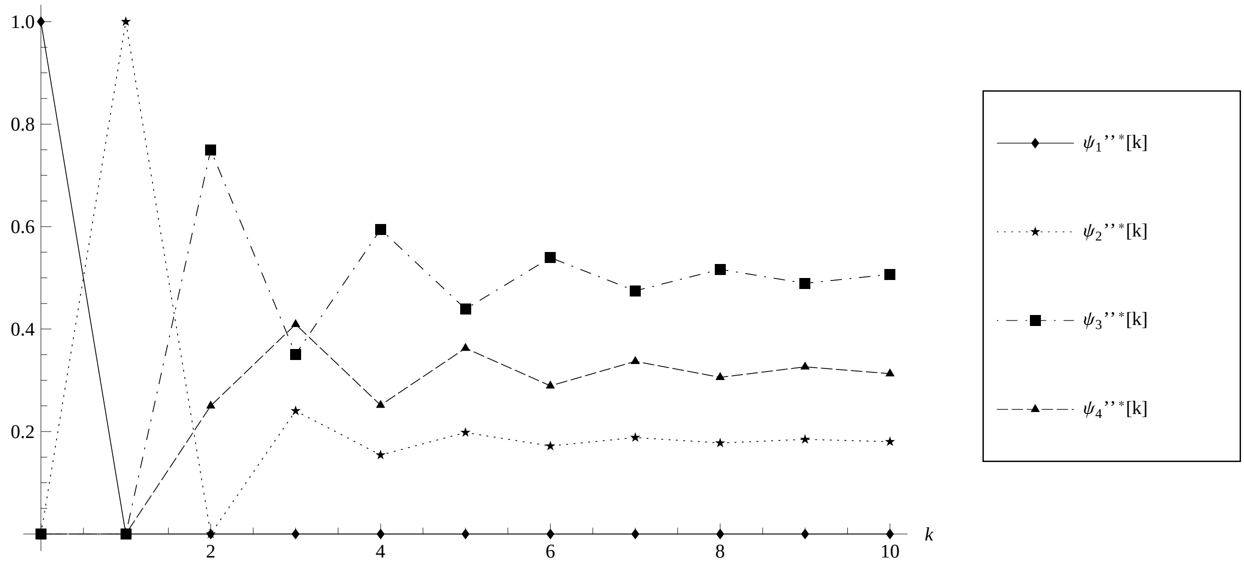

Figure 47: Transient probabilities alteration diagram of the quotient abstract dining philosophers system

Only one philosopher eats in the state $\mathcal{K}_3$. Then, the *fraction of time when only one philosopher dines* is $\psi_3''^* = \frac{1}{2}$.

Two philosophers eat together in the state $\mathcal{K}_4$. Then, the *fraction of time when two philosophers dine* is $\psi_4''^* = \frac{7}{22}$.

The *relative fraction of time when two philosophers dine with respect to when only one philosopher dines* is $\frac{7}{22} \cdot \frac{2}{1} = \frac{7}{11}$.

- The beginning of eating of a philosopher $\{b\}$ is only possible from the states $\mathcal{K}_2, \mathcal{K}_3$. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of multiactions containing $\{b\}$. Thus, the *steady-state probability of the beginning of eating of a philosopher* is $\psi_2''^* \sum_{\{A,\mathcal{K}|\{b\}\in A,\ \mathcal{K}_2 \xrightarrow{A} \mathcal{K}\}} PM_A^*(\mathcal{K}_2,\mathcal{K}) + \psi_3''^* \sum_{\{A,\mathcal{K}|\{b\}\in A,\ \mathcal{K}_3 \xrightarrow{A} \mathcal{K}\}} PM_A^*(\mathcal{K}_3,\mathcal{K}) = \frac{2}{11}\left(\frac{3}{4}+\frac{1}{4}\right) + \frac{1}{2}\left(\frac{6}{11}+\frac{2}{11}\right) = \frac{6}{11}$.

One can see that the performance indices are the same for the complete and the quotient abstract dining philosophers systems. The explanation of this fact is just the same as that presented earlier for the complete and the reduced abstract dining philosophers systems.

10.2.3 The generalized system

Let us determine which is the influence of the multiaction probabilities from specification of the dining philosophers system on its performance. Suppose that all the mentioned multiactions have the same generalized probability $\rho \in (0;1)$. The resulting specification K of the generalized dining philosophers system is defined as follows.

The static expression of the philosopher i $(1 \le i \le 4)$ is $K_i = [(\{x_i\},\rho) * (((\{b_i,\widehat{y_i}\},\rho);(\{e_i,\widehat{z_i}\},\rho))[]((\{y_{i+1}\},\rho);(\{z_{i+1}\},\rho))) * \mathsf{Stop}]$. The static expression of the philosopher 5 is $K_5 = [(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\rho) * (((\{b_5,\widehat{y_5}\},\rho);(\{e_5,\widehat{z_5}\},\rho))[]((\{y_1\},\rho);(\{z_1\},\rho))) * \mathsf{Stop}]$. The static expression of the generalized dining philosophers system is $K = (K_1\|K_2\|K_3\|K_4\|K_5)$ sy x_1 sy x_2 sy x_3 sy x_4 sy y_1 sy y_2 sy y_3 sy y_4 sy y_5 sy z_1 sy z_2 sy z_3 sy z_4 sy z_5 rs x_1 rs x_2 rs x_3 rs x_4 rs y_1 rs y_2 rs y_3 rs y_4 rs y_5 rs z_1 rs z_2 rs z_3 rs z_4 rs z_5.

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

The TPM for $DTMC^*(\overline{K})$ is

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

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

$$\tilde{\psi}^* = \left(0, \frac{1}{2(3-\rho^2)}, \frac{1}{10}, \frac{2-\rho^2}{10(3-\rho^2)}, \frac{2-\rho^2}{10(3-\rho^2)}, \frac{1}{10}, \frac{1}{10}, \frac{2-\rho^2}{10(3-\rho^2)}, \frac{2-\rho^2}{10(3-\rho^2)}, \frac{1}{10}, \frac{1}{10}, \frac{2-\rho^2}{10(3-\rho^2)}\right).$$

We can now calculate the main performance indices.

- The average recurrence time in the state $\tilde{s}_2$, where all the forks are available, called the *average system run-through*, is $\frac{1}{\tilde{\psi}_2^*} = 2(3-\rho^2)$.
- Nobody eats in the state $\tilde{s}_2$. Then, the *fraction of time when no philosophers dine* is $\tilde{\psi}_2^* = \frac{1}{2(3-\rho^2)}$.

  Only one philosopher eats in the states $\tilde{s}_3, \tilde{s}_6, \tilde{s}_7, \tilde{s}_{10}, \tilde{s}_{11}$. Then, the *fraction of time when only one philosopher dines* is $\tilde{\psi}_3^* + \tilde{\psi}_6^* + \tilde{\psi}_7^* + \tilde{\psi}_{10}^* + \tilde{\psi}_{11}^* = \frac{1}{10} + \frac{1}{10} + \frac{1}{10} + \frac{1}{10} + \frac{1}{10} = \frac{1}{2}$.

  Two philosophers eat together in the states $\tilde{s}_4, \tilde{s}_5, \tilde{s}_8, \tilde{s}_9, \tilde{s}_{12}$. Then, the *fraction of time when two philosophers dine* is $\tilde{\psi}_4^* + \tilde{\psi}_5^* + \tilde{\psi}_8^* + \tilde{\psi}_9^* + \tilde{\psi}_{12}^* = \frac{2-\rho^2}{10(3-\rho^2)} + \frac{2-\rho^2}{10(3-\rho^2)} + \frac{2-\rho^2}{10(3-\rho^2)} + \frac{2-\rho^2}{10(3-\rho^2)} + \frac{2-\rho^2}{10(3-\rho^2)} = \frac{2-\rho^2}{2(3-\rho^2)}$.

  The *relative fraction of time when two philosophers dine with respect to when only one philosopher dines* is $\frac{2-\rho^2}{2(3-\rho^2)} \cdot \frac{2}{1} = \frac{2-\rho^2}{3-\rho^2}$.
- The beginning of eating of first philosopher $(\{b_1\}, \rho^2)$ is only possible from the states $\tilde{s}_2, \tilde{s}_6, \tilde{s}_7$. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of activities containing $(\{b_1\}, \rho^2)$. Thus, the *steady-state probability of the beginning of eating of first philosopher* is $\tilde{\psi}_2^* \sum_{\{\Gamma \mid (\{b_1\},\rho^2)\in\Gamma\}} PT^*(\Gamma, \tilde{s}_2) + \tilde{\psi}_6^* \sum_{\{\Gamma \mid (\{b_1\},\rho^2)\in\Gamma\}} PT^*(\Gamma, \tilde{s}_6) + \tilde{\psi}_7^* \sum_{\{\Gamma \mid (\{b_1\},\rho^2)\in\Gamma\}} PT^*(\Gamma, \tilde{s}_7) = \frac{1}{2(3-\rho^2)}\left(\frac{1-\rho^2}{5} + \frac{\rho^2}{5} + \frac{\rho^2}{5}\right) + \frac{1}{10}\left(\frac{1-\rho^2}{3-\rho^2} + \frac{\rho^2}{3-\rho^2}\right) + \frac{1}{10}\left(\frac{1-\rho^2}{3-\rho^2} + \frac{\rho^2}{3-\rho^2}\right) = \frac{3+\rho^2}{10(3-\rho^2)}$.

10.2.4 The abstract generalized system and its reductions

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

The static expression of the philosopher i $(1 \le i \le 4)$ is $L_i = [(\{x_i\}, \rho) * (((\{b, \widehat{y_i}\}, \rho); (\{e, \widehat{z_i}\}, \rho))[]((\{y_{i+1}\}, \rho); (\{z_{i+1}\}, \rho))) * \mathsf{Stop}]$. The static expression of the philosopher 5 is $L_5 = [(\{a, \widehat{x_1}, \widehat{x_2}, \widehat{x_2}, \widehat{x_4}\}, \rho) * (((\{b, \widehat{y_5}\}, \rho); (\{e, \widehat{z_5}\}, \rho))[]((\{y_1\}, \rho); (\{z_1\}, \rho))) * \mathsf{Stop}]$. The static expression of the abstract generalized dining philosophers system is $L = (L_1 \| L_2 \| L_3 \| L_4 \| L_5)$ sy x_1 sy x_2 sy x_3 sy x_4 sy y_1 sy y_2 sy y_3 sy y_4 sy y_5 sy z_1 sy z_2 sy z_3 sy z_4 sy z_5 rs x_1 rs x_2 rs x_3 rs x_4 rs y_1 rs y_2 rs y_3 rs y_4 rs y_5 rs z_1 rs z_2 rs z_3 rs z_4 rs z_5.

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

The first performance index and the second group of the indices are the same for the generalized system and its abstract modification. Let us consider the following performance index based on non-personalized viewpoint to the philosophers.

- The beginning of eating of a philosopher $(\{b\}, \rho^2)$ is only possible from the states $\tilde{s}_2, \tilde{s}_3, \tilde{s}_6, \tilde{s}_7, \tilde{s}_{10}, \tilde{s}_{11}$. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all

multisets of activities containing $(\{b\},\rho^2)$. Thus, the *steady-state probability of the beginning of eating of a philosopher* is
$\tilde{\psi}_2^* \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT^*(\Gamma,\tilde{s}_2) + \tilde{\psi}_3^* \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT^*(\Gamma,\tilde{s}_3) + \tilde{\psi}_6^* \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT^*(\Gamma,\tilde{s}_6) +$
$\tilde{\psi}_7^* \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT^*(\Gamma,\tilde{s}_7) + \tilde{\psi}_{10}^* \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT^*(\Gamma,\tilde{s}_{10}) + \tilde{\psi}_{11}^* \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT^*(\Gamma,\tilde{s}_{11}) =$
$\frac{1}{2(3-\rho^2)}\left(\frac{1-\rho^2}{5}+\frac{\rho^2}{5}+\frac{1-\rho^2}{5}+\frac{\rho^2}{5}+\frac{1-\rho^2}{5}+\frac{\rho^2}{5}+\frac{1-\rho^2}{5}+\frac{\rho^2}{5}+\frac{1-\rho^2}{5}+\frac{\rho^2}{5}\right)+$
$\frac{1}{10}\left(\frac{1-\rho^2}{3-\rho^2}+\frac{\rho^2}{3-\rho^2}+\frac{1-\rho^2}{3-\rho^2}+\frac{\rho^2}{3-\rho^2}\right)+\frac{1}{10}\left(\frac{1-\rho^2}{3-\rho^2}+\frac{\rho^2}{3-\rho^2}+\frac{1-\rho^2}{3-\rho^2}+\frac{\rho^2}{3-\rho^2}\right)+\frac{1}{10}\left(\frac{1-\rho^2}{3-\rho^2}+\frac{\rho^2}{3-\rho^2}+\frac{1-\rho^2}{3-\rho^2}+\frac{\rho^2}{3-\rho^2}\right)+$
$\frac{1}{10}\left(\frac{1-\rho^2}{3-\rho^2}+\frac{\rho^2}{3-\rho^2}+\frac{1-\rho^2}{3-\rho^2}+\frac{\rho^2}{3-\rho^2}\right)+\frac{1}{10}\left(\frac{1-\rho^2}{3-\rho^2}+\frac{\rho^2}{3-\rho^2}+\frac{1-\rho^2}{3-\rho^2}+\frac{\rho^2}{3-\rho^2}\right)=\frac{3}{2(3-\rho^2)}$.

Let us consider a reduction of the abstract generalized dining philosophers system. The static expression of the philosopher 1 is $L_1' = [(\{x\},\rho) * ((\{b\},\frac{2\rho^2}{1+\rho^2});(\{e\},\rho^2)) * \mathsf{Stop}]$. The static expression of the philosopher 2 is $L_2' = [(\{a,\hat{x}\},\rho^4) * ((\{b\},\frac{2\rho^2}{1+\rho^2});(\{e\},\rho^2)) * \mathsf{Stop}]$. The static expression of the reduced abstract generalized dining philosophers system is $L' = (L_1' \| L_2')$ sy x rs x.

$DR(\overline{L'})$ consists of the 5 states which are interpreted as follows: $\tilde{s}_1'$ is the initial state, $\tilde{s}_2'$: the system is activated and no philosophers dine, $\tilde{s}_3', \tilde{s}_4'$: one philosopher dines, $\tilde{s}_5'$: two philosophers dine.

Consider the equivalence $\mathcal{R} : \overline{L} \underline{\leftrightarrow}_{ss} \overline{L'}$ such that $(DR(\overline{L}) \cup DR(\overline{L'}))/_{\mathcal{R}} = \{\widetilde{\mathcal{H}}_1, \widetilde{\mathcal{H}}_2, \widetilde{\mathcal{H}}_3, \widetilde{\mathcal{H}}_4\}$, where $\widetilde{\mathcal{H}}_1 = \{\tilde{s}_1, \tilde{s}_1'\}$ (the initial state), $\widetilde{\mathcal{H}}_2 = \{\tilde{s}_2, \tilde{s}_2'\}$ (the system is activated and no philosophers dine), $\widetilde{\mathcal{H}}_3 = \{\tilde{s}_3, \tilde{s}_6, \tilde{s}_7, \tilde{s}_{10}, \tilde{s}_{11}, \tilde{s}_3', \tilde{s}_4'\}$ (one philosopher dines), $\widetilde{\mathcal{H}}_4 = \{\tilde{s}_4, \tilde{s}_5, \tilde{s}_8, \tilde{s}_9, \tilde{s}_{12}, \tilde{s}_5'\}$ (two philosophers dine). One can see that L' is a reduction of L with respect to $\underline{\leftrightarrow}_{ss}$.

The TPM for $DTMC^*(\overline{L'})$ is

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

The steady-state PMF for $DTMC^*(\overline{L'})$ is

$$\tilde{\psi}'^* = \left(0, \frac{1}{2(3-\rho^2)}, \frac{1}{4}, \frac{1}{4}, \frac{2-\rho^2}{2(3-\rho^2)}\right).$$

We can now calculate the main performance indices.

- The average recurrence time in the state $\tilde{s}_2'$, where all the forks are available, called the *average system run-through*, is $\frac{1}{\tilde{\psi}_2'^*} = 2(3-\rho^2)$.

- Nobody eats in the state $\tilde{s}_2'$. Then, the *fraction of time when no philosophers dine* is $\tilde{\psi}_2'^* = \frac{1}{2(3-\rho^2)}$.

  Only one philosopher eats in the states $\tilde{s}_3', \tilde{s}_4'$. Then, the *fraction of time when only one philosopher dines* is $\tilde{\psi}_3'^* + \tilde{\psi}_4'^* = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$.

  Two philosophers eat together in the state $\tilde{s}_5'$. Then, the *fraction of time when two philosophers dine* is $\tilde{\psi}_5'^* = \frac{2-\rho^2}{2(3-\rho^2)}$.

  The *relative fraction of time when two philosophers dine with respect to when only one philosopher dines* is $\frac{2-\rho^2}{2(3-\rho^2)} \cdot \frac{2}{1} = \frac{2-\rho^2}{3-\rho^2}$.

- The beginning of eating of a philosopher $(\{b\},\frac{2\rho^2}{1+\rho^2})$ is only possible from the states $\tilde{s}_2', \tilde{s}_3', \tilde{s}_4'$. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of activities containing $(\{b\},\frac{2\rho^2}{1+\rho^2})$. Thus, the *steady-state probability of the beginning of eating of a philosopher* is $\tilde{\psi}_2'^* \sum_{\{\Gamma|(\{b\},\frac{2\rho^2}{1+\rho^2})\in\Gamma\}} PT^*(\Gamma,\tilde{s}_2') + \tilde{\psi}_3'^* \sum_{\{\Gamma|(\{b\},\frac{2\rho^2}{1+\rho^2})\in\Gamma\}} PT^*(\Gamma,\tilde{s}_3') + \tilde{\psi}_4'^* \sum_{\{\Gamma|(\{b\},\frac{2\rho^2}{1+\rho^2})\in\Gamma\}} PT^*(\Gamma,\tilde{s}_4') =$
  $\frac{1}{2(3-\rho^2)}\left(\frac{1-\rho^2}{2}+\frac{1-\rho^2}{2}+\rho^2\right)+\frac{1}{4}\left(\frac{2(1-\rho^2)}{3-\rho^2}+\frac{2\rho^2}{3-\rho^2}\right)+\frac{1}{4}\left(\frac{2(1-\rho^2)}{3-\rho^2}+\frac{2\rho^2}{3-\rho^2}\right)=\frac{3}{2(3-\rho^2)}$.

One can see that the performance indices are the same for the complete and the reduced abstract generalized dining philosophers systems. The coincidence of the first performance index, as well as the second group of indices obviously illustrates the result of Proposition 8.1. The coincidence of the third performance index is due to Theorem 8.2: one should just apply its result to the step traces $\{\{b\}\}$, $\{\{b\},\{b\}\}$, $\{\{b\},\{e\}\}$ of the expressions $\overline{L}$ and $\overline{L'}$, and then sum the left and right parts of the three resulting equalities.

Note that $TS^*(\overline{L'})$ can be reduced further by merging the equivalent states $\tilde{s}'_3$ and $\tilde{s}'_4$, thus, it can be transformed into a transition system with four states only. But the resulting reduction of the initial transition system $TS^*(\overline{L})$ will not correspond to some *dtsPBC* expression anymore.

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

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

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

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

$$\tilde{\psi}''^{*} = \left(0, \frac{1}{2(3-\rho^2)}, \frac{1}{2}, \frac{2-\rho^2}{2(3-\rho^2)}\right).$$

We can now calculate the main performance indices.

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

  Only one philosopher eats in the state $\widetilde{\mathcal{K}}_3$. Then, the *fraction of time when only one philosopher dines* is $\tilde{\psi}_3''^{*} = \frac{1}{2}$.

  Two philosophers eat together in the state $\widetilde{\mathcal{K}}_4$. Then, the *fraction of time when two philosophers dine* is $\tilde{\psi}_4''^{*} = \frac{2-\rho^2}{2(3-\rho^2)}$.

  The *relative fraction of time when two philosophers dine with respect to when only one philosopher dines* is $\frac{2-\rho^2}{2(3-\rho^2)} \cdot \frac{2}{1} = \frac{2-\rho^2}{3-\rho^2}$.
- The beginning of eating of a philosopher $\{b\}$ is only possible from the states $\widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}_3$. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of multiactions containing $\{b\}$. Thus, the *steady-state probability of the beginning of eating of a philosopher* is $\tilde{\psi}_2''^{*} \sum_{\{A,\widetilde{\mathcal{K}} \mid \{b\} \in A,\ \widetilde{\mathcal{K}}_2 \stackrel{A}{\twoheadrightarrow} \widetilde{\mathcal{K}}\}} PM_A^*(\widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}) + \tilde{\psi}_3''^{*} \sum_{\{A,\widetilde{\mathcal{K}} \mid \{b\} \in A,\ \widetilde{\mathcal{K}}_3 \stackrel{A}{\twoheadrightarrow} \widetilde{\mathcal{K}}\}} PM_A^*(\widetilde{\mathcal{K}}_3, \widetilde{\mathcal{K}}) = \frac{1}{2(3-\rho^2)}((1-\rho^2)+\rho^2) + \frac{1}{2}\left(\frac{2(1-\rho^2)}{3-\rho^2} + \frac{2\rho^2}{3-\rho^2}\right) = \frac{3}{2(3-\rho^2)}$.

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

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

The steady-state probabilities $\tilde{\psi}_1''^{*} = 0$ and $\tilde{\psi}_3''^{*} = \frac{1}{2}$ are constants. Therefore, only $\tilde{\psi}_2''^{*} = \frac{1}{2(3-\rho^2)}$ and $\tilde{\psi}_4''^{*} = \frac{2-\rho^2}{2(3-\rho^2)}$ depend on ρ. Note that $\tilde{\psi}_2''^{*} + \tilde{\psi}_4''^{*} = \frac{1}{2(3-\rho^2)} + \frac{2-\rho^2}{2(3-\rho^2)} = \frac{1}{2}$, hence, the sum of these steady-state probabilities does not depend on ρ. This fact has the interpretation in terms of performance indices: the fraction of time when no or two philosophers dine coincides with that when only one philosopher dines, and both fractions are equal to $\frac{1}{2}$.

In Figure 48, the plots of $\tilde{\psi}_2''^{*}$ and $\tilde{\psi}_4''^{*}$ as functions of ρ are depicted. The diagrams are symmetric with respect to the constant probability $\frac{1}{4}$. One can see that, the more is value of ρ, the less is difference between $\tilde{\psi}_2''^{*}$ and $\tilde{\psi}_4''^{*}$. Since $\tilde{\psi}_4''^{*} - \tilde{\psi}_2''^{*} = \frac{2-\rho^2}{2(3-\rho^2)} - \frac{1}{2(3-\rho^2)} = \frac{1-\rho^2}{2(3-\rho^2)}$, the difference tends to $\frac{1}{6}$ when ρ approaches 0, whereas it tends to 0 when ρ approaches 1. Notice that, however, we do not allow $\rho = 0$ or $\rho = 1$. The difference can be treated as that between the fractions of time when two and when no philosophers dine.

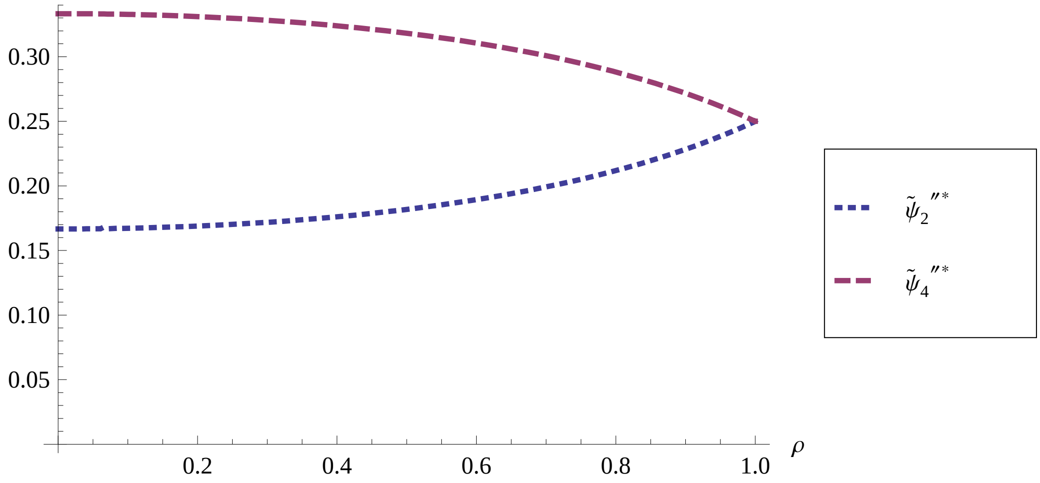

Figure 48: Steady-state probabilities $\tilde{\psi}_2''^*$ and $\tilde{\psi}_4''^*$ as functions of the parameter ρ

From the performance viewpoint, it is more interesting is to consider the expression $\tilde{\psi}_3''^* + \tilde{\psi}_4''^* - \tilde{\psi}_2''^* = \frac{1}{2} + \frac{2-\rho^2}{2(3-\rho^2)} - \frac{1}{2(3-\rho^2)} = \frac{2-\rho^2}{3-\rho^2}$. Its value tends to $\frac{2}{3}$ when ρ approaches 0, whereas it tends to $\frac{1}{2}$ when ρ approaches 1. The value can be interpreted as the difference between the fractions of time when some (one or two) and when no philosophers dine. Thus, when ρ is closer to 0, more time is spent for eating and less time remains for thinking, i.e. *dining is preferred.* When ρ is closer to 1, the situation is symmetric, i.e. *thinking is preferred.*

The influence of value ρ to the main performance indices presented before is investigated according to the same pattern as above.

11 Conclusion

In this paper, we have considered a discrete time stochastic extension $dtsPBC$ of a finite part of PBC enriched with iteration. The calculus has a concurrent step operational semantics based on transition systems and a denotational semantics in terms of a subclass of LDTSPNs. Within the context of $dtsPBC$ with iteration, we have defined a number of stochastic algebraic equivalences which have natural net analogues on LDTSPNs. The equivalences abstract from empty loops in transition systems corresponding to dynamic expressions. The diagram of interrelations for the algebraic equivalences has been constructed. We have explained how one can reduce transition systems and DTMCs, as well as expressions and dts-boxes modulo the stochastic equivalences. We have presented a logical characterization of the stochastic bisimulation equivalences. An application of the equivalences to comparison of the stationary behaviour has been demonstrated, and we have found which of the equivalences we proposed guarantee identity of the stationary behaviour in the equivalence classes. We have proved that the weakest of the relations having this property is the step stochastic bisimulation equivalence. A congruence relation has been proposed.

A method of modeling, performance evaluation and performance preserving reduction of concurrent stochastic systems was proposed based on steady-state probabilities analysis. The transition systems and underlying DTMCs of expressions were reduced w.r.t. step stochastic bisimulation equivalence that guarantees identity of the stationary behaviour and thus preserves performance measures. The method was applied to the shared memory system and dining philosophers system, as well as to their generalized versions with a variable probability of activities. This probability was interpreted as a parameter of the performance index functions. The influence of the parameter value to the systems' performance was analyzed with the goal of optimization.

The advantage of our framework is twofold. First, one can specify in it concurrent composition and synchronization of (multi)actions, whereas this is not possible in classical Markov chains. Second, algebraic formulas represent processes in a more compact way than Petri nets and allow one to apply syntactic transformations and comparisons. Process algebras are compositional by definition and their operations naturally correspond to operators of programming languages. Hence, it is much easier to construct a complex model in the algebraic setting than in PNs. The complexity of PNs generated for practical models in the literature demonstrates that it is not straightforward to construct such PNs directly from the system specifications. $dtsPBC$ is well suited for the discrete time applications, whose discrete states change with a global time tick, such as business processes, neural and transportation networks, computer and communication systems, timed web services [150], as well as for those, in which the distributed architecture or the concurrency level should be preserved while modeling and analysis (remember that, in step semantics, we have additional transitions due to concurrent executions).

Future work consists in abstracting from the silent activities in the definitions of the equivalences, i.e. from the activities with empty multiaction part. The abstraction from empty loops and that from silent activities could be done in one step as well. The main point here is that we should collect probabilities during such abstractions from an internal activity. As a result, we shall have the algebraic analogues of the net stochastic equivalences from [36, 37]. Moreover, we plan to extend *dtsPBC* with recursion to enhance specification power of the calculus.

References

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

[2] Andova S. *Process algebra with probabilistic choice. Lecture Notes in Computer Science* **1601**, p. 111–129, 1999.

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

[4] Bravetti M., D'Argenio P.R. *Tutte le algebre insieme: concepts, discussions and relations of stochastic process algebras with general distributions. Lecture Notes in Computer Science* **2925**, p. 44–88, 2004, `http://www.cs.unibo.it/~bravetti/papers/voss03.ps`.

[5] Baier C. *Polynomial time algorithms for testing probabilistic bisimulation and simulation. Lecture Notes in Computer Science* **1102**, p. 50–61, 1996, `http://www.inf.tu-dresden.de/content/institutes/thi/algi/publikationen/texte/27_00_old.pdf`.

[6] Balbo G. *Introduction to stochastic Petri nets. Lecture Notes in Computer Science* **2090**, p. 84–155, 2001.

[7] Balbo G. *Introduction to generalized stochastic Petri nets. Lecture Notes in Computer Science* **4486**, p. 83–131, 2007.

[8] Baarir S., Beccuti M., Dutheillet C., Franceschinis G., Haddad S. *Lumping partially symmetric stochastic models. Performance Evaluation* **68**, p. 21–44, 2011.

[9] Bravetti M., Bernardo M., Gorrieri R. *Towards performance evaluation with general distributions in process algebras. Lecture Notes in Computer Science* **1466**, p. 405–422, September 1998, `http://www.cs.unibo.it/~bravetti/papers/concur98.ps`.

[10] Best E., Devillers R., Esparza J. *General refinement and recursion operations in the box calculus. Lecture Notes in Computer Science* **665**, p. 130–140, 1993.

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

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

[13] Best E., Devillers R., Koutny M. *Petri nets, process algebras and concurrent programming languages. Lecture Notes in Computer Science* **1492**, p. 1–84, 1998, `http://parsys.informatik.uni-oldenburg.de/~best/publications/apnf.ps.gz`.

[14] Best E., Devillers R., Koutny M. *The box algebra: a model of nets and process expressions. Lecture Notes in Computer Science* **1639**, p. 344–363, 1999.

[15] Best E., Devillers R., Koutny M. *Petri net algebra. EATCS Monographs on Theoretical Computer Science*, 378 p., Springer Verlag, 2001.

[16] Best E., Devillers R., Koutny M. *A unified model for nets and process algebras.* In: *Handbook of Process Algebra*, Chapter **14**, p. 873–944, (Bergstra J.A., Ponse A., Smolka S.A., eds.), Elsevier Science B.V., Amsterdam, The Netherlands, 2001, `http://parsys.informatik.uni-oldenburg.de/~best/publications/handbook.ps.gz`.

[17] Best E., Devillers R., Koutny M. *Recursion and Petri nets. Acta Informatica* **37(11–12)**, p. 781–829, 2001, `http://parsys.informatik.uni-oldenburg.de/~best/publications/bdk-recursion.ps.gz`.

[18] Best E., Devillers R., Koutny M. *The box algebra = Petri nets + process expressions. Information and Computation* **178**, p. 44–100, 2002.

[19] Baier C., Engelen B., Majster-Cederbaum M. *Deciding bisimilarity and similarity for probabilistic processes. Journal of Computer and System Sciences* **60(1)**, p. 187–231, Elsevier, February 2000.

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

[21] Bernardo M. *A survey of Markovian behavioral equivalences. Lecture Notes in Computer Science* **4486**, p. 180–219, 2007, `http://www.sti.uniurb.it/bernardo/documents/sfm07pe.pdf`.

[22] Best E. *COSY and its relationship to CSP. Lecture Notes in Computer Science* **255**, p. 416–440, 1986.

[23] Bernardo M., Gorrieri R. *A tutorial on EMPA: a theory of concurrent processes with nondeterminism, priorities, probabilities and time. Theoretical Computer Science* **202**, p. 1–54, July 1998, `http://www.sti.uniurb.it/bernardo/documents/tcs202.pdf`.

[24] Baier C., Hermanns H. *Weak bisimulation for fully probabilistic processes. Lecture Notes in Computer Science* **1254**, p. 119–130, 1997, `http://web.informatik.uni-bonn.de/l/baier/papers/BH97.ps`.

[25] Baier C., Hermanns H., Katoen J.-P., Wolf V. *Comparative branching-time semantics for Markov chains. Lecture Notes in Computer Science* **2761**, p. 492–507, 2003, `http://www.inf.tu-dresden.de/content/institutes/thi/algi/publikationen/texte/11_00.pdf`.

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

[27] Buchholz P., Kemper P. *Quantifying the dynamic behavior of process algebras. Lecture Notes in Computer Science* **2165**, p. 184–199, 2001.

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

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

[30] Best E., Koutny M. *Solving recursive net equations. Lecture Notes in Computer Science* **944**, p. 605–623, 1995.

[31] Bause F., Kritzinger P.S. *Stochastic Petri nets: an introduction to the theory.* Vieweg Verlag, 2^{nd} edition, 218 p., 2002, `http://ls4-www.cs.tu-dortmund.de/cms/de/home/bause/bause_kritzinger_spn_book_print.pdf`.

[32] Best E., Linde-Göers H.G. *Compositional process semantics of Petri boxes. Lecture Notes in Computer Science* **802**, p. 250–270, 1993.

[33] Bloom B., Meyer A. *A remark on bisimulation between probabilistic processes. Lecture Notes in Computer Science* **363**, p. 26–40, 1989.

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

[35] Brinksma E., Hermanns H. *Process algebra and Markov chains. Lecture Notes in Computer Science* **2090**, p. 183–231, 2001.

[36] Buchholz P., Tarasyuk I.V. *A class of stochastic Petri nets with step semantics and related equivalence notions. Technische Berichte* **TUD-FI00-12**, 18 p., Fakultät Informatik, Technische Universität Dresden, Germany, November 2000, `ftp://ftp.inf.tu-dresden.de/pub/berichte/tud00-12.ps.gz`.

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

[38] Buchholz P., Tarasyuk I.V. *Equivalences for stochastic Petri nets and stochastic process algebras. Vestnik, Quartal Journal of Novosibirsk State University, Series: Mathematics, Mechanics and Informatics* **6(1)**, p. 14–42, Novosibirsk State University, Novosibirsk, 2006 (in Russian), `http://itar.iis.nsk.su/files/itar/pages/vestnik06.pdf`.

[39] Buchholz P. *Markovian process algebra: composition and equivalence.* In: Herzog U. and Rettelbach M., editors, *Proceedings of* 2^{nd} *Workshop on Process Algebras and Performance Modelling*, *Arbeitsberichte des IMMD* **27**, p. 11–30, University of Erlangen, Germany, 1994.

[40] Buchholz P. *Equivalence relations for stochastic automata networks.* In: W.J. Stewart, ed., *Computation with Markov Chains*, p. 197–216, Kluwer Academic Publishers, USA, 1995.

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

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

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

[44] Cattani S., Segala R. *Decision algorithms for probabilistic bisimulation. Lecture Notes in Computer Science* **2421**, p. 371–385, 2002, `http://www.cs.bham.ac.uk/~dxp/papers/CS02.pdf`, `http://qav.cs.ox.ac.uk/papers/CS02.pdf`.

[45] Daws C. *Symbolic and parametric model checking of discrete-time Markov chains. Proceedings of* 1^{st} *International Colloquium on Theoretical Aspects of Computing - 04 (ICTAC'04), Lecture Notes in Computer Science* **3407**, p. 280–294, 2005.

[46] Devillers R. *Construction of S-invariants and S-components for refined Petri boxes. Lecture Notes in Computer Science* **691**, p. 242–261, 1993.

[47] Devillers R. *S-invariant analysis of general recursive Petri boxes. Acta Informatica* **32(4)**, p. 313–345, 1995.

[48] Devillers R. *Petri boxes and finite processes. Lecture Notes in Computer Science* **1119**, p. 465–480, 1996.

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

[50] Degano P., Priami C. *Non-interleaving semantics for mobile processes. Theoretical Computer Science* **216(1–2)**, p. 237–270, March 1999.

[51] Donatelli S., Ribaudo M., Hillston J. *A comparison of performance evaluation process algebra and generalized stochastic Petri nets. Proceedings of* 6^{th} *International Workshop on Petri Nets and Performance Models*, Durham, USA, IEEE Computer Society Press, p. 158–168, 1995.

[52] Esparza J., Bruns G. *Trapping mutual exclusion in the box calculus. Theoretical Computer Science* **153**, p. 95–128, 1996.

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

[54] de Frutos E.D., Marroquín A.O. *Ambient Petri nets. Electronic Notes in Theoretical Computer Science* **85(1)**, 27 p., 2003, `http://www.elsevier.nl/gej-ng/31/29/23/138/47/27/85.1.005.ps`, `http://www.elsevier.nl/locate/entcs/volume85.html`.

[55] Florin G., Natkin S. *Les reseaux de Petri stochastiques. Technique et Science Informatique* **4(1)**, 1985.

[56] FOURNEAU J.M. *Collaboration of discrete-time Markov chains: Tensor and product form. Performance Evaluation* **67**, p. 779–796, 2010.

[57] GILMORE S., HILLSTON J., KLOUL L., RIBAUDO M. *PEPA nets: a structured performance modelling formalism. Performance Evaluation* **54**, p. 79–104, 2003, `http://www.dcs.ed.ac.uk/pepa/pepanetsJournal.pdf`.

[58] VAN GLABBEEK R.J. *The linear time – branching time spectrum II: the semantics of sequential systems with silent moves. Extended abstract. Lecture Notes in Computer Science* **715**, p. 66–81, 1993.

[59] VAN GLABBEEK R.J., SMOLKA S.A., STEFFEN B. *Reactive, generative, and stratified models of probabilistic processes. Information and Computation* **121(1)**, p. 59–80, 1995, `http://boole.stanford.edu/pub/prob.ps.gz`.

[60] HANSSON H. *Time and probability in formal design of distributed systems.* In: *Real-Time Safety Critical Systems*, Volume **1**, Elsevier, The Netherlands, 1994.

[61] HAVERKORT B.R. *Markovian models for performance and dependability evaluation. Lecture Notes in Computer Science* **2090**, p. 38–83, 2001, `http://www-i2.informatik.rwth-aachen.de/Teaching/Seminar/VOSS2005/have01.pdf`.

[62] HERZOG U. *Formal methods for performance evaluation. Lecture Notes in Computer Science* **2090**, p. 1–37, 2001.

[63] HILLSTON J. *The nature of synchronisation. Proceedings of the* 2^{nd} *International Workshop on Process Algebra and Performance Modelling - 94 (PAPM'94)*, Regensberg / Erlangen (U. Herzog, M. Rettelbach, eds.), *Arbeitsberichte des IMMD* **27**, p. 51–70, University of Erlangen, Germany, November 1994, `http://www.dcs.ed.ac.uk/pepa/synchronisation.pdf`.

[64] HILLSTON J. *A compositional approach to performance modelling.* 158 p., Cambridge University Press, UK, 1996, `http://www.dcs.ed.ac.uk/pepa/book.pdf`.

[65] HAN T., KATOEN J.-P., MEREACRE A. *Approximate parameter synthesis for probabilistic time-bounded reachability. Proceedings of* 29^{th} *IEEE Real-Time Systems Symposium - 08 (RTSS'08)*, p. 173–182, IEEE Computer Society Press, New York, USA, 2008.

[66] HESKETH M., KOUTNY M. *An axiomatization of duplication equivalence in the Petri box calculus. Lecture Notes in Computer Science* **1420**, p. 165–184, 1998.

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

[68] HORVÁTH A., PAOLIERI M., RIDI L., VICARIO E. *Probabilistic model checking of non-Markovian models with concurrent generally distributed timers. Proceedings of* 8^{th} *International Conference on the Quantitative Evaluation of Systems - 11 (QEST'11)*, p. 131–140, Aachen, Germany, September 2011, `http://www.di.unito.it/~horvath/publications/papers/HoPaRiVi11.pdf`.

[69] HORVÁTH A., PAOLIERI M., RIDI L., VICARIO E. *Transient analysis of non-Markovian models using stochastic state classes. Performance Evaluation* **69(7–8)**, p. 315–335, 2012.

[70] HERMANNS H., RETTELBACH M. *Syntax, semantics, equivalences and axioms for MTIPP. Proceedings of* 2^{nd} *Workshop on Process Algebras and Performance Modelling*, Regensberg / Erlangen (Herzog U., Rettelbach M., eds.), *Arbeitsberichte des IMMD* **27**, p. 71–88, University of Erlangen, Germany, 1994, `http://ftp.informatik.uni-erlangen.de/local/inf7/papers/Hermanns/syntax_semantics_equivalences_axioms_for_MTIPP.ps.gz`.

[71] JOU C.-C., SMOLKA S.A. *Equivalences, congruences and complete axiomatizations for probabilistic processes. Lecture Notes in Computer Science* **458**, p. 367–383, 1990.

[72] JATEGAONKAR L., MEYER A.R. *Deciding true concurrency equivalences on safe, finite nets. Theoretical Computer Science* **154(1)**, p. 107–143, 1996.

[73] JONSSON B., YI W., LARSEN K.G. *Probabilistic extensions of process algebras.* In: *Handbook of Process Algebra*, Chapter **11**, p. 685–710, (Bergstra J.A., Ponse A., Smolka S.A., eds.), Elsevier Science B.V., Amsterdam, The Netherlands, 2001.

[74] Katoen J.-P., D'Argenio P.R. *General distributions in process algebra. Lecture Notes in Computer Science* **2090**, p. 375–429, 2001.

[75] Kahn G. *Natural semantics. Proceedings of* 4^{th} *Annual Symposium on Theoretical Aspects of Computer Science - 87 (STACS'87)*, February 1987, p. 22–39, Springer, London, UK, 1987.

[76] Koutny M., Best E. *Operational and denotational semantics for the box algebra. Theoretical Computer Science* **211(1–2)**, p. 1–83, 1999, `http://parsys.informatik.uni-oldenburg.de/~best/publications/tcs.ps.gz`.

[77] Katoen J.-P., Brinksma E., Latella D., Langerak R. *Stochastic simulation of event structures. Proceedings of* 4^{th} *International Workshop on Process Algebra and Performance Modelling - 96 (PAPM'96)* (M. Ribaudo, ed.), p. 21–40, CLUT Press, Torino, Italy, July 1996, `http://eprints.eemcs.utwente.nl/6487/01/263_KLLB96b.pdf`.

[78] Kotov V.E., Cherkasova L.A. *On structural properties of generalized processes. Lecture Notes in Computer Science* **188**, p. 288–306, 1985.

[79] Koutny M., Esparza J., Best E. *Operational semantics for the Petri box calculus. Lecture Notes in Computer Science* **836**, p. 210–225, 1994.

[80] Kwiatkowska M.Z., Norman G.J., Parker D. *Symmetry reduction for probabilistic model checking. Lecture Notes in Computer Science* **4144**, p. 234–248, 2006.

[81] Kotov V.E. *An algebra for parallelism based on Petri nets. Lecture Notes in Computer Science* **64**, p. 39–55, 1978.

[82] Koutny M. *Partial order semantics of box expressions. Lecture Notes in Computer Science* **815**, p. 318–337, 1994.

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

[84] Kulkarni V.G. *Modeling and analysis of stochastic systems. Texts in Statistical Science*, 542 p., Chapman and Hall / CRC Press, 2010.

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

[86] Lanotte R., Maggiolo-Schettini A., Troina A. *Parametric probabilistic transition systems for system design and analysis. Formal Aspects of Computing* **19**, p. 93–109, 2007.

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

[88] Larsen K.G., Skou A. *Compositional verification of probabilistic processes. Lecture Notes in Computer Science* **630**, p. 456–471, 1992.

[89] Markovski J., D'Argenio P.R., Baeten J.C.M., de Vink E.P. *Reconciling real and stochastic time: the need for probabilistic refinement. Formal Aspects of Computing* **24(4–6)**, p. 497–518, 2012.

[90] Macià H.S. *sPBC: Una extensión Markoviana del Petri box calculus. Ph.D. thesis*, 249 p., Departamento de Informática, Universidad de Castilla - La Mancha, Albacete, Spain, December 2003 (in Spanish), `http://www.info-ab.uclm.es/retics/publications/2003/sPBCthesis03.pdf`.

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

[92] Marsan M.A., Balbo G., Conte G., Donatelli S., Franceschinis G. *Modelling with generalized stochastic Petri nets. Wiley Series in Parallel Computing*, John Wiley and Sons, 316 p., 1995, `http://www.di.unito.it/~greatspn/GSPN-Wiley/`.

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

[94] Marroquín A.O., de Frutos E.D. *Extending the Petri box calculus with time. Lecture Notes in Computer Science* **2075**, p. 303–322, 2001.

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

[96] Milner R.A.J. *Communication and concurrency.* Prentice-Hall, 260 p., Upper Saddle River, NJ, USA, 1989.

[97] Molloy M.K. *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.

[98] Molloy M.K. *Performance analysis using stochastic Petri nets. IEEE Transactions on Software Engineering* **31(9)**, p. 913–917, 1982.

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

[100] Milner R.A.J., Parrow J.G., Walker D.J. *A calculus of mobile processes, I. Information and Computation* **100(1)**, p. 1–40, September 1992.

[101] Milner R.A.J., Parrow J.G., Walker D.J. *A calculus of mobile processes, II. Information and Computation* **100(1)**, p. 41–77, September 1992.

[102] Montanari U., Pistore M., Yankelevich D. *Efficient minimization up to location equivalence. Lecture Notes in Computer Science* **1058**, p. 265–279, 1996.

[103] Macià S.H., Valero R.V., Cuartero G.F. *A congruence relation in finite sPBC. Technical Report* **DIAB-02-01-31**, 34 p., Department of Computer Science, University of Castilla - La Mancha, Albacete, Spain, October 2002, `http://www.info-ab.uclm.es/retics/publications/2002/tr020131.ps`.

[104] Macià S.H., Valero R.V., Cuartero G.F. *Defining equivalence relations in sPBC. Proceedings of* 1^{st} *International Conference on the Principles of Software Engineering - 04 (PriSE'04)*, p. 195–205, Buenos Aires, Argentina, November 2004, `http://www.info-ab.uclm.es/retics/publications/2004/prise04.pdf`.

[105] Macià S.H., Valero R.V., Cazorla L.D., Cuartero G.F. *Introducing the iteration in sPBC. Technical Report* **DIAB-03-01-37**, 20 p., Department of Computer Science, University of Castilla - La Mancha, Albacete, Spain, September 2003, `http://www.info-ab.uclm.es/descargas/tecnicalreports/DIAB-03-01-37/diab030137.zip`.

[106] Macià S.H., Valero R.V., Cazorla L.D., Cuartero G.F. *Introducing the iteration in sPBC. Proceedings of the* 24^{th} *International Conference on Formal Techniques for Networked and Distributed Systems - 04 (FORTE'04)*, Madrid, Spain, *Lecture Notes in Computer Science* **3235**, p. 292–308, October 2004, `http://www.info-ab.uclm.es/retics/publications/2004/forte04.pdf`.

[107] Macià S.H., Valero R.V., Cuartero G.F., de Frutos E.D. *A congruence relation for sPBC. Formal Methods in System Design* **32(2)**, p. 85–128, Springer, The Netherlands, April 2008.

[108] Macià S.H., Valero R.V., Cuartero G.F., Pelayo L.F. *A new proposal for the synchronization in sPBC. Technical Report* **DIAB-02-01-26**, 15 p., Department of Computer Science, University of Castilla - La Mancha, Albacete, Spain, June 2002, `http://www.info-ab.uclm.es/sec-ab/Tecrep/newproposalsysPBC.ps`.

[109] Macià S.H., Valero R.V., Cuartero G.F., Pelayo L.F. *Improving the synchronization in stochastic Petri box calculus. Actas de las II Jornadas sobre Programacion y Lenguajes - 02 (PROLE'02)*, El Escorial, Spain, November 2002.

[110] Macià S.H., Valero R.V., Cuartero G.F., Pelayo L.F. *A new synchronization in finite stochastic Petri box calculus. Proceedings of* 3^{rd} *International IEEE Conference on Application of Concurrency to System Design - 03 (ACSD'03)*, p. 216–225, Guimarães, Portugal, IEEE Computer Society Press, June 2003, `http://www.info-ab.uclm.es/retics/publications/2003/acsd03.pdf`.

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

[112] Macià S.H., Valero R.V., de Frutos E.D. *sPBC: a Markovian extension of finite Petri box calculus. Proceedings of 9^{th} IEEE International Workshop on Petri Nets and Performance Models - 01 (PNPM'01)*, p. 207–216, Aachen, Germany, IEEE Computer Society Press, September 2001, `http://www.info-ab.uclm.es/retics/publications/2001/pnpm01.ps`.

[113] Macià S.H., Valero R.V., de Frutos E.D. *sPBC: a Markovian extension of finite PBC. Actas de IX Jornadas de Concurrencia - 01 (JC'01)*, p. 243–256, Sitges, Spain, June 2001, `http://www.info-ab.uclm.es/retics/publications/2001/mvfjc01.ps`.

[114] Macià S.H., Valero R.V., de Frutos E.D., Cuartero G.F. *Extending PBC with Markovian multiactions. Proceedings of XXVII Conferencia Latinoamericana de Informática - 01 (CLEI'01)* (Montilva, J.A., Besembel, I., eds.), 12 p., Mérida, Venezuela, Universidad de los Andes, September 2001, `http://www.info-ab.uclm.es/retics/publications/2001/clei01.ps`.

[115] Markovski J., de Vink E.P. *Extending timed process algebra with discrete stochastic time. Proceedings of 12^{th} International Conference on Algebraic Methodology and Software Technology - 08 (AMAST'08)*, Urbana, IL, USA, *Lecture Notes of Computer Science* **5140**, p. 268–283, 2008.

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

[117] Núñez G.M., de Frutos E.D., Llana D.L. *Acceptance trees for probabilistic processes. Lecture Notes in Computer Science* **962**, p. 249–263, 1995.

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

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

[120] Núñez G.M. *An axiomatization of probabilistic testing. Lecture Notes in Computer Science* **1601**, p. 130–150, 1999, `http://dalila.sip.ucm.es/miembros/manolo/papers/arts99.ps.gz`.

[121] Peterson J.L. *Petri net theory and modeling of systems.* Prentice-Hall, 1981.

[122] Plotkin G.D. *A structural approach to operational semantics. Technical Report* **DAIMI FN-19**, Computer Science Department, Aarhus University, Aarhus, Denmark, 1981.

[123] Priami C. *Stochastic π-calculus with general distributions. Proceedings of 4^{th} International Workshop on Process Algebra and Performance Modelling - 96 (PAPM'96)* (M. Ribaudo, ed.), p. 41–57, CLUT Press, Torino, Italy, 1996.

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

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

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

[127] Ribaudo M. *Stochastic Petri net semantics for stochastic process algebra. Proceedings of 6^{th} International Workshop on Petri Nets and Performance Models*, p. 148–157, Durham, NC, USA, IEEE Computer Society Press, 1995.

[128] Tarasyuk I.V. *Equivalence notions applied to designing concurrent systems with the use of Petri nets. Programming and Computer Software* **24(4)**, p. 162–175, Nauka, Moscow, 1998, `http://www.maik.rssi.ru/journals/procom.htm`.

[129] Tarasyuk I.V. *Logical characterization of probabilistic τ-bisimulation equivalences. Joint Novosibirsk Computing Center and Institute of Informatics Systems Bulletin, Series Computer Science* **20**, p. 97–111, Novosibirsk, 2004, `http://itar.iis.nsk.su/files/itar/pages/prlogncc.pdf`.

[130] Tarasyuk I.V. *Stochastic Petri nets: a formalism for modeling and performance analysis of computing processes. System Informatics* **9**, p. 135–194, SB RAS Publisher, Novosibirsk, 2004 (in Russian), http://itar.iis.nsk.su/files/itar/pages/spnsi.pdf.

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

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

[133] Tarasyuk I.V. *Stochastic Petri box calculus with discrete time. Fundamenta Informaticae* **76(1–2)**, p. 189–218, IOS Press, Amsterdam, The Netherlands, February 2007, http://itar.iis.nsk.su/files/itar/pages/dtspbcfi.pdf.

[134] Tarasyuk I.V. *Equivalences for behavioural analysis of concurrent and distributed computing systems.* 321 p., Academic Publisher "Geo", Novosibirsk, 2007 (ISBN 978-5-9747-0098-9, in Russian).

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

[136] Tarasyuk I.V. *A notion of congruence for dtsPBC. Bulletin of the Novosibirsk Computing Center, Series Computer Science, IIS Special Issue* **28**, p. 121–141, NCC Publisher, Novosibirsk, 2008, http://itar.iis.nsk.su/files/itar/pages/dtcgncc.pdf.

[137] Tarasyuk I.V. *Performance evaluation in dtsPBC. Proceedings of* 18^{th} *Workshop on Concurrency, Specification and Programming - 09 (CS&P'09)*, Kraków-Przegorzały, Poland, September 28–30, 2009, L. Czaja, M. Szczuka, editors, p. 602–615, Warsaw University, 2009, http://itar.iis.nsk.su/files/itar/pages/dtsshmcsp.pdf.

[138] Tarasyuk I.V. *Modeling and performance analysis of concurrent processes in the algebra dtsPBC. Vestnik, Quartal Journal of Novosibirsk State University, Series: Mathematics, Mechanics, Informatics* **9(4)**, p. 90–117, Novosibirsk State University, Novosibirsk, 2009 (in Russian), http://itar.iis.nsk.su/files/itar/pages/vestnik09.pdf.

[139] Tarasyuk I.V. *Equivalence relations for behaviour-preserving reduction and modular performance evaluation in dtsPBC. Berichte aus dem Department für Informatik* **01/10**, 75 p., Carl von Ossietzky Universität Oldenburg, Germany, April 2010, http://itar.iis.nsk.su/files/itar/pages/dtspbcpeib_cov.pdf.

[140] Tarasyuk I.V. *Performance preserving equivalences for dtsPBC. Bulletin of the Novosibirsk Computing Center, Series Computer Science, IIS Special Issue* **31**, p. 155–178, NCC Publisher, Novosibirsk, 2010, http://itar.iis.nsk.su/files/itar/pages/dtspencc.pdf.

[141] Tarasyuk I.V. *Performance analysis of the dining philosophers system in dtsPBC. Pre-proceedings of* 8^{th} *Ershov Informatics Conference - 11 (PSI'11)*, 17 p., Novosibirsk, Russia, 2011, http://itar.iis.nsk.su/files/itar/pages/dtsdphpsi.pdf.

[142] Tarasyuk I.V. *Equivalences for modular performance analysis in dtsPBC. Berichte aus dem Department für Informatik* **04/11**, 41 p., Carl von Ossietzky Universität Oldenburg, Germany, October 2011, http://itar.iis.nsk.su/files/itar/pages/dtsdphib_cov.pdf.

[143] Tarasyuk I.V. *Performance evaluation of the generalized shared memory system in dtsPBC. Bulletin of the Novosibirsk Computing Center, Series Computer Science, IIS Special Issue* **32**, p. 127–155, NCC Publisher, Novosibirsk, 2011, http://itar.iis.nsk.su/files/itar/pages/dtsgsmncc.pdf.

[144] Tarasyuk I.V. *Equivalence relations for modular performance evaluation in dtsPBC. Mathematical Structures in Computer Science* **24(1)**, p. 78–154 (e240103), Cambridge University Press, Cambridge, UK, February 2014, http://itar.iis.nsk.su/files/itar/pages/dtsdphms.pdf.

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

[146] Tarasyuk I.V., Macià S.H., Valero R.V. *Stochastic equivalence for modular performance evaluation in dtsiPBC. Technical Report* **DIAB-11-06-2**, 50 p., Department of Computer Systems, High School of Computer Science Engineering, University of Castilla - La Mancha, Albacete, Spain, June 2011, `http://itar.iis.nsk.su/files/itar/pages/dtsipbceq.pdf`, `http://www.dsi.uclm.es/descargas/technicalreports/DIAB-11-06-2/report_dtsipbdeq.pdf`.

[147] Tarasyuk I.V., Macià S.H., Valero R.V. *Discrete time stochastic Petri box calculus with immediate multiactions. Pre-proceedings of* 6^{th} *International Workshop on Practical Applications of Stochastic Modelling - 12 (PASM'12)*, 21 p., Imperial College London, UK, September 2012, `http://itar.iis.nsk.su/files/itar/pages/dtsipbcpasm.pdf`.

[148] Tarasyuk I.V., Macià S.H., Valero R.V. *Applying stochastic equivalence to performance evaluation in dtsiPBC. Technical Report* **DIAB-12-10-2**, 62 p., Department of Computer Systems, High School of Computer Science Engineering, University of Castilla - La Mancha, Albacete, Spain, October 2012, `http://itar.iis.nsk.su/files/itar/pages/dtsipbcpe.pdf`, `http://www.dsi.uclm.es/descargas/technicalreports/DIAB-12-10-2/technicalreport_dtsiPBC_oct2012.pdf`.

[149] Tarasyuk I.V., Macià S.H., Valero R.V. *Discrete time stochastic Petri box calculus with immediate multiactions dtsiPBC. Proceedings of* 6^{th} *International Workshop on Practical Applications of Stochastic Modelling - 12 (PASM'12) and* 11^{th} *International Workshop on Parallel and Distributed Methods in Verification - 12 (PDMC'12), Electronic Notes in Theoretical Computer Science*, 25 p., Elsevier, August 2013 (ISSN 1571-0661), `http://itar.iis.nsk.su/files/itar/pages/dtsipbcentcs.pdf`.

[150] Valero R.V., Cambronero P.M.E. *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)**, p. 570–594, Tailor and Francis, 2017.

[151] Wolf V., Baier C., Majster-Cederbaum M. *Trace machines for observing continuous-time Markov chains. Proceedings of the* 3^{rd} *International Workshop on Quantitative Aspects of Programming Languages (QAPL'05)*, Edinburgh, UK, 2005, *Electronic Notes in Theoretical Computer Science* **153(2)**, p. 259–277, 2006.

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

A Proofs

A.1 Proof of Theorem 4.2

At some points, the present proof for *dtsPBC* goes along the lines from the respective proofs for *PBC* [13–16, 29, 76].

We first give a necessary definition. Let $(P_N, T_N, W_N, \Omega_N, \mathcal{L}_N, M_N)$ be an LDTSPN. A set of transitions $U \subseteq Ena(M)$ is *fireable* at a marking $M \in I\!N_{fin}^{P_N}$, if ${}^\bullet U \subseteq M$ and $Ena(M) \subseteq Ts_N$. In other words, a set of transitions U is fireable at a marking M, if it has enough tokens in its input places at M. Let $Fire(M)$ be the set *all transition sets fireable at* M. By the definition of fireability, it follows that $Fire(M) \subseteq 2^{T_N}$. A transition $t \in Ena(M)$ is *fireable* at a marking M, denoted by $t \in Fire(M)$, if $\{t\} \in Fire(M)$. A transition $t \in Fire(M)$ fires with probability $\Omega_N(t)$ when no different transition is fireable at M, i.e. $Fire(M) = \{\emptyset, \{t\}\}$. By the definition of fireability, $\forall U \in Fire(M)\ 2^U \setminus \{\emptyset\} \subseteq Fire(M)$.

Let $N = Box_{dts}(E)$. We define a mapping $\beta : DR(\overline{E}) \to RS(\overline{N})$ so that $\beta([G]_\approx) = M_G$ iff $[G]_\approx \in DR(\overline{E})$ and $(N, M_G) = Box_{dts}(G)$. Then, like in *PBC* [13–16, 29, 76], one can see that β is a bijection, since for each dynamic expression G its structural equivalence class $[G]_\approx$ defines a single corresponding marking M_G in the dts-box $Box_{dts}(G)$ and vice versa.

Clearly, $[\overline{E}]_\approx \in DR(\overline{E})$ and $Box_{dts}(\overline{E}) = \overline{Box_{dts}(E)} = \overline{N} = (N, {}^\circ N) = (N, M_{\overline{E}})$. Hence, $\beta([\overline{E}]_\approx) = M_{\overline{E}}$. Thus, β binds the initial states of the transition system $TS(\overline{E})$ and the corresponding reachability graph $RG(\overline{N})$.

Let $[G]_\approx \in DR(\overline{E})$ and $\beta([G]_\approx) = M_G \in RS(\overline{N})$. We now prove by induction on the structure of dynamic expressions and corresponding dts-boxes that $Exec([G]_\approx)$ and $Fire(M_G)$ are isomorphic. This means that for every $\Gamma \in Exec([G]_\approx)$ there exists $U \in Fire(M_G)$ such that U consists of the transitions *corresponding* to the activities from Γ and vice versa: $(\alpha, \rho)_\iota \in \Gamma \ \Leftrightarrow\ t_\iota \in U$, where $\Lambda_N(t_\iota) = \varrho_{(\alpha,\rho)}$. Thus, the *corresponding* activities and transitions have the same probabilities, as well as the same multiaction labels and numberings. We can write $U = U(\Gamma)$ and $\Gamma = \Gamma(U)$, to indicate such a correspondence.

Actually, each Γ and the *corresponding* U are completely defined by the sets of their numberings $Num(\Gamma) = \{\iota \mid (\alpha,\rho)_\iota \in \Gamma\} = \{\iota \mid t_\iota \in U\} = Num(U)$, since each activity and transition have a unique numbering. Moreover, $Exec([G]_\approx)$ and $Fire(M_G)$ are completely defined by the sets of their numberings $Num(Exec([G]_\approx)) = \{Num(\Gamma) \mid \Gamma \in Exec([G]_\approx)\} = \{Num(U) \mid U \in Fire(M_G)\} = Num(Fire(M_G))$.

- If $final(G)$ then $G \approx \underline{E}$ and $Exec([G]_\approx) = Exec([\underline{E}]_\approx) = \{\emptyset\}$. On the other hand, $Box_{dts}(G) = Box_{dts}(\underline{E}) = \underline{N} = (N, N^\circ) = (N, M_{\underline{E}})$ and $Fire(M_G) = Fire(M_{\underline{E}}) = \{\emptyset\} = Exec([G]_\approx)$.

- If $G = \overline{(\alpha,\rho)_\iota}$ then $Exec([G]_\approx) = \{\emptyset, \{(\alpha,\rho)_\iota\}\}$. On the other hand, $Box_{dts}(G) = (N_{(\alpha,\rho)_\iota}, {}^\bullet t_\iota)$, where $\Lambda_N(t_\iota) = \varrho_{(\alpha,\rho)}$, and $Fire(M_G) = Fire({}^\bullet t_\iota) = \{\emptyset, \{t_\iota\}\}$, which is isomorphic to $Exec([G]_\approx)$.

- If $G = H; E$, where $H \in OpRegDynExpr,\ E \in RegStatExpr$, then

$$Exec([H;E]_\approx) = \begin{cases} Exec([H]_\approx), & \neg final(H); \\ Exec([\overline{E}]_\approx) & final(H). \end{cases}$$

  On the other hand, $Box_{dts}(G) = Box_{dts}(H;E) = (Box_{dts}(\lfloor H \rfloor; E), M_{H;E})$, and for $Box_{dts}(H) = (Box_{dts}(\lfloor H \rfloor), M_H),\ Box_{dts}(\overline{E}) = \overline{N_E} = (N_E, {}^\circ N_E) = (N_E, M_{\overline{E}})$, we have

$$Fire(M_{H;E}) = \begin{cases} Fire(M_H), & M_H \neq N_H^\circ; \\ Fire(M_{\overline{E}}), & M_H = N_H^\circ; \end{cases}$$

  which is isomorphic to $Exec([H;E]_\approx)$.

- If $G = E; H$, where $E \in RegStatExpr,\ H \in OpRegDynExpr$, then

$$Exec([E;H]_\approx) = Exec([H]_\approx).$$

  On the other hand, $Box_{dts}(G) = Box_{dts}(E;H) = (Box_{dts}(E; \lfloor H \rfloor), M_{E;H})$, and for $Box_{dts}(H) = (Box_{dts}(\lfloor H \rfloor), M_H)$, we have

$$Fire(M_{E;H}) = Fire(M_H);$$

  which is isomorphic to $Exec([E;H]_\approx)$.

- If $G = H[]E$, where $H \in OpRegDynExpr,\ E \in RegStatExpr$, then

$$Exec([H[]E]_\approx) = \begin{cases} Exec([H]_\approx), & \neg init(H); \\ Exec([H]_\approx) \cup Exec([\overline{E}]_\approx), & init(H). \end{cases}$$

  On the other hand, $Box_{dts}(G) = Box_{dts}(H[]E) = (Box_{dts}(\lfloor H \rfloor []E), M_{H[]E})$, and for $Box_{dts}(H) = (Box_{dts}(\lfloor H \rfloor), M_H),\ Box_{dts}(\overline{E}) = \overline{N_E} = (N_E, {}^\circ N_E) = (N_E, M_{\overline{E}})$, we have

$$Fire(M_{H[]E}) = \begin{cases} Fire(M_H), & M_H \neq {}^\circ N_H; \\ Fire(M_H) \cup Fire(M_{\overline{E}}), & M_H = {}^\circ N_H; \end{cases}$$

  which is isomorphic to $Exec([H[]E]_\approx)$.

  If $G = E[]H$, where $E \in RegStatExpr,\ H \in OpRegDynExpr$, then the constructions are similar.

- If $G = H\|Z$, where $H, Z \in OpRegDynExpr$, then

$$Exec([H\|Z]_\approx) = Exec([H]_\approx) \cup Exec([Z]_\approx) \cup (Exec([H]_\approx) \odot Exec([Z]_\approx)),$$

  where $Exec([H]_\approx) \odot Exec([Z]_\approx) = \{\Gamma + \Delta \mid \Gamma \in Exec([H]_\approx),\ \Delta \in Exec([Z]_\approx)\}$.

  On the other hand, $Box_{dts}(G) = Box_{dts}(H\|Z) = (Box_{dts}(\lfloor H \rfloor \| Z), M_{H\|Z})$, and for $Box_{dts}(H) = (Box_{dts}(\lfloor H \rfloor), M_H),\ Box_{dts}(Z) = (Box_{dts}(\lfloor Z \rfloor), M_Z)$, we have

$$Fire(M_{H\|Z}) = Fire(M_H) \cup Fire(M_Z) \cup (Fire(M_H) \odot Fire(M_Z)),$$

  where $Fire(M_H) \odot Fire(M_Z) = \{U \cup T \mid U \in Fire(M_H),\ T \in Fire(M_Z)\}$; which is isomorphic to $Exec([H\|Z]_\approx)$.

- If $G = H[f]$, where $H \in \mathit{OpRegDynExpr}$, then

$$\mathit{Exec}([H[f]]_{\approx}) = \{f(\Gamma) \mid \Gamma \in \mathit{Exec}([H]_{\approx})\}.$$

On the other hand, $\mathit{Box}_{dts}(G) = \mathit{Box}_{dts}(H[f]) = (\mathit{Box}_{dts}(\lfloor H \rfloor[f]), M_{H[f]})$, and for $\mathit{Box}_{dts}(H) = (\mathit{Box}_{dts}(\lfloor H \rfloor), M_H)$, we have

$$\mathit{Fire}(M_{H[f]}) = \{f(U) \mid U \in \mathit{Fire}(M_H)\},$$

where $f(U) = \{t_\iota \in U \mid \Lambda_H(t_\iota) = \varrho_{(\alpha,\rho)},\ \Lambda_{H[f]}(t_\iota) = \varrho_{(f(\alpha),\rho)}\}$; which is isomorphic to $\mathit{Exec}([H[f]]_{\approx})$.

- If $G = H \text{ rs } a$, where $H \in \mathit{OpRegDynExpr}$, then

$$\mathit{Exec}([H \text{ rs } a]_{\approx}) = \{\Gamma - \Gamma_a \mid \Gamma \in \mathit{Exec}([H]_{\approx})\},$$

where $\Gamma_a = \{(\alpha, \rho)_\iota \in \Gamma \mid (a \in \alpha) \vee (\hat{a} \in \alpha)\},\ a \in \mathit{Act}$.
On the other hand, $\mathit{Box}_{dts}(G) = \mathit{Box}_{dts}(H \text{ rs } a) = (\mathit{Box}_{dts}(\lfloor H \rfloor \text{ rs } a), M_{H \text{ rs } a})$, and for $\mathit{Box}_{dts}(H) = (\mathit{Box}_{dts}(\lfloor H \rfloor), M_H)$, we have

$$\mathit{Fire}(M_{H \text{ rs } a}) = \{U \setminus U_a \mid U \in \mathit{Fire}(M_H)\},$$

where $U_a = \{t_\iota \in U \mid \Lambda_H(t_\iota) = \varrho_{(\alpha,\rho)},\ (a \in \alpha) \vee (\hat{a} \in \alpha)\},\ a \in \mathit{Act}$; which is isomorphic to $\mathit{Exec}([H \text{ rs } a]_{\approx})$.

- If $G = H \text{ sy } a$, where $H \in \mathit{OpRegDynExpr}$, then

$$\mathit{Exec}([H \text{ sy } a]_{\approx}) = \begin{array}{l}\mathit{Exec}([H]_{\approx}) \cup \{\Gamma + \{(\alpha \oplus_a \beta, \rho \cdot \chi)_{(\iota_1)(\iota_2)}\} \mid \Gamma + \{(\alpha, \rho)_{\iota_1}\} + \{(\beta, \chi)_{\iota_2}\} \in \mathit{Exec}([H]_{\approx}),\\ a \in \alpha,\ \hat{a} \in \beta\}.\end{array}$$

On the other hand, $\mathit{Box}_{dts}(G) = \mathit{Box}_{dts}(H \text{ sy } a) = (\mathit{Box}_{dts}(\lfloor H \rfloor \text{ sy } a), M_{H \text{ sy } a})$, and for $\mathit{Box}_{dts}(H) = (\mathit{Box}_{dts}(\lfloor H \rfloor), M_H)$, we have

$$\mathit{Fire}(M_{H \text{ sy } a}) = \begin{array}{l}\mathit{Fire}(M_H) \cup \{U \cup \{t_{(\iota_1)(\iota_2)}\} \mid \Lambda_{H \text{ sy } a}(t_{(\iota_1)(\iota_2)}) = \varrho_{(\alpha \oplus_a \beta, \rho \cdot \chi)},\ U \cup \{v_{\iota_1}, w_{\iota_2}\} \in \mathit{Fire}(M_H),\\ \Lambda_H(v_{\iota_1}) = \varrho_{(\alpha,\rho)},\ \Lambda_H(w_{\iota_2}) = \varrho_{(\beta,\chi)},\ a \in \alpha,\ \hat{a} \in \beta\};\end{array}$$

which is isomorphic to $\mathit{Exec}([H \text{ sy } a]_{\approx})$.

- If $G = [H * E * F]$, where $H \in \mathit{OpRegDynExpr},\ E, F \in \mathit{RegStatExpr}$, then

$$\mathit{Exec}([[H * E * F]]_{\approx}) = \begin{cases} \mathit{Exec}([H]_{\approx}), & \neg \mathit{final}(H);\\ \mathit{Exec}([\overline{E}]_{\approx}) \cup \mathit{Exec}([\overline{F}]_{\approx}), & \mathit{final}(H).\end{cases}$$

On the other hand, $\mathit{Box}_{dts}(G) = \mathit{Box}_{dts}([H * E * F]) = (\mathit{Box}_{dts}(\lfloor H \rfloor * E * F), M_{[H*E*F]})$, and for $\mathit{Box}_{dts}(H) =$
$(\mathit{Box}_{dts}(\lfloor H \rfloor), M_H),\ \mathit{Box}_{dts}(\overline{E}) = \overline{N_E} = (N_E, {}^\circ N_E) = (N_E, M_{\overline{E}}),\ \mathit{Box}_{dts}(\overline{F}) = \overline{N_F} = (N_F, {}^\circ N_F) = (N_F, M_{\overline{F}})$, we have

$$\mathit{Fire}(M_{[H*E*F]}) = \begin{cases} \mathit{Fire}(M_H), & M_H \neq N_H^\circ;\\ \mathit{Fire}(M_{\overline{E}}) \cup \mathit{Fire}(M_{\overline{F}}), & M_H = N_H^\circ;\end{cases}$$

which is isomorphic to $\mathit{Exec}([[H * E * F]]_{\approx})$.

- If $G = [E * H * F]$, where $E, F \in \mathit{RegStatExpr},\ H \in \mathit{OpRegDynExpr}$, then

$$\mathit{Exec}([[E * H * F]]_{\approx}) = \begin{cases} \mathit{Exec}([H]_{\approx}), & \neg \mathit{init}(H) \wedge \neg \mathit{final}(H);\\ \mathit{Exec}([H]_{\approx}) \cup \mathit{Exec}([\overline{F}]_{\approx}), & \mathit{init}(H) \vee \mathit{final}(H).\end{cases}$$

On the other hand, $\mathit{Box}_{dts}(G) = \mathit{Box}_{dts}([E * H * F]) = (\mathit{Box}_{dts}(E * \lfloor H \rfloor * F), M_{[E*H*F]})$, and for $\mathit{Box}_{dts}(H) =$
$(\mathit{Box}_{dts}(\lfloor H \rfloor), M_H),\ \mathit{Box}_{dts}(\overline{F}) = \overline{N_F} = (N_F, {}^\circ N_F) = (N_F, M_{\overline{F}})$, we have

$$\mathit{Fire}(M_{[E*H*F]}) = \begin{cases} \mathit{Fire}(M_H), & M_H \neq {}^\circ N_H \wedge M_H \neq N_H^\circ;\\ \mathit{Fire}(M_H) \cup \mathit{Fire}(M_{\overline{F}}), & M_H = {}^\circ N_H \vee M_H = N_H^\circ;\end{cases}$$

which is isomorphic to $\mathit{Exec}([[E * H * F]]_{\approx})$.

- If $G = [E * F * H]$, where $E, F \in RegStatExpr$, $H \in OpRegDynExpr$, then

$$Exec([[E * F * H]]_\approx) = \begin{cases} Exec([H]_\approx), & \neg init(H); \\ Exec([\overline{F}]_\approx) \cup Exec([H]_\approx), & init(H). \end{cases}$$

On the other hand, $Box_{dts}(G) = Box_{dts}([E * F * H]) = (Box_{dts}(E * F * \lfloor H \rfloor), M_{[E*F*H]})$, and for $Box_{dts}(\overline{F}) =$
$\overline{N_F} = (N_F, {}^\circ N_F) = (N_F, M_{\overline{F}})$, $Box_{dts}(H) = (Box_{dts}(\lfloor H \rfloor), M_H)$, we have

$$Fire(M_{[E*F*H]}) = \begin{cases} Fire(M_H), & M_H \neq {}^\circ N_H; \\ Fire(M_{\overline{F}}) \cup Fire(M_H), & M_H = {}^\circ N_H; \end{cases}$$

which is isomorphic to $Exec([[E * F * H]]_\approx)$.

Thus, we have proved that $Exec([G]_\approx)$ and $Fire(M_G)$ are isomorphic. It remains to check the homomorphism property, stating that for all $[G]_\approx, [\widetilde{G}]_\approx \in DR(\overline{E})$ and for all *corresponding* $\Gamma \in Exec([G]_\approx)$, $U \in Fire(M_G)$ it holds $[G]_\approx \xrightarrow{\Gamma}_\mathcal{P} [\widetilde{G}]_\approx \Leftrightarrow M_G = \beta([G]_\approx) \xrightarrow{U}_\mathcal{P} \beta([\widetilde{G}]_\approx) = M_{\widetilde{G}}$.

The probability functions $PF(\Gamma, [G]_\approx)$ and $PT(\Gamma, [G]_\approx)$ depend only on the structure of $Exec([G]_\approx)$, as well as on the probabilities of stochastic multiactions from its elements. Analogously, $PF(U, M_G)$ and $PT(U, M_G)$ depend only on the structure of $Fire(M_G)$, as well as the probabilities of stochastic transitions from its elements. Further, $PF(\Gamma, [G]_\approx)$ and $PT(\Gamma, [G]_\approx)$ are respectively defined in the same way (using the same formulas and cases) as $PF(U, M_G)$ and $PT(U, M_G)$, for each pair of the *corresponding* (multi)set of activities Γ and transition set U. Obviously, the isomorphism of $Exec([G]_\approx)$ and $Fire(M_G)$ guarantees coincidence of their structure as well as the mentioned probabilities and weights. Hence, if U *corresponds* to Γ then $PF(\Gamma, [G]_\approx) = PF(U, M_G)$ and $PT(\Gamma, [G]_\approx) = PT(U, M_G)$.

We also have $\mathcal{L}(\Gamma) = \mathcal{L}(U)$, where $\mathcal{L}(U) = \sum_{\{t \in U \mid \Lambda_G(t) = \varrho_{(\alpha,\rho)}\}} \alpha$ is the *multiaction part* of a set of transitions $U \subseteq T_N$. Thus, each transition $[G]_\approx \xrightarrow{\Gamma}_\mathcal{P} \tilde{s}$ in $TS(\overline{E})$ has a corresponding one $M_G \xrightarrow{U}_\mathcal{P} \widetilde{M}$ in $RG(\overline{N})$ with $\mathcal{L}(\Gamma) = \mathcal{L}(U)$ and vice versa. Observe that the structure of the plain and operator dts-boxes in *dtsPBC* is similar to that of the plain and operator boxes in *PBC*. Hence, like in *PBC* [13–16, 29, 76], we can prove that $\tilde{s} = [\widetilde{G}]_\approx$ and $\widetilde{M} = M_{\widetilde{G}}$ with $(N, M_{\widetilde{G}}) = Box_{dts}(\widetilde{G})$ for the dynamic expression $\widetilde{G}$ such that $G \xrightarrow{\Gamma} \widetilde{G}$. Therefore, by construction of β, we get $\beta([\widetilde{G}]_\approx) = M_{\widetilde{G}}$. □

A.2 Proof of Proposition 5.2

It is enough to prove the statement for $\star = s$, since the case $\star = i$ is considered analogously.

Like it has been done for strong equivalence in Proposition 8.2.1 from [64], we shall prove the following fact about step stochastic bisimulation. Let us have $\forall j \in \mathcal{J}\ \mathcal{R}_j : G \underline{\leftrightarrow}_{ss} G'$ for some index set $\mathcal{J}$. Then the transitive closure of the union of all relations $\mathcal{R} = (\cup_{j \in \mathcal{J}} \mathcal{R}_j)^+$ is also an equivalence and $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$.

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

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

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

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

It remains to prove that $(s_1, s_2) \in \mathcal{R}$ implies $\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_\mathcal{R}\ \forall A \in \mathbb{N}^\mathcal{L}_{fin} \setminus \{\emptyset\}\ PM^*_A(s_1, \mathcal{H}) = PM^*_A(s_2, \mathcal{H})$.

- $n = 1$

  In this case, $(s_1, s_2) \in \mathcal{R}$ implies $\exists j \in \mathcal{J}\ (s_1, s_2) \in \mathcal{R}_j$. Since $\mathcal{R}_j : G \underline{\leftrightarrow}_{ss} G'$, we get $\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_\mathcal{R}\ \forall A \in \mathbb{N}^\mathcal{L}_{fin} \setminus \{\emptyset\}$

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

- $n \to n + 1$

  Suppose that $\forall m \leq n\ (s_1, s_2) \in \mathcal{R}(m)$ implies $\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_\mathcal{R}\ \forall A \in \mathbb{N}^\mathcal{L}_{fin} \setminus \{\emptyset\}\ PM^*_A(s_1, \mathcal{H}) = PM^*_A(s_2, \mathcal{H})$.

Then $(s_1, s_2) \in \mathcal{R}(n+1)$ implies $\exists j \in \mathcal{J}\ (s_1, s_2) \in \mathcal{R}_j \circ \mathcal{R}(n)$, i.e. $\exists s_3 \in (DR(G) \cup DR(G'))$ such that $(s_1, s_3) \in \mathcal{R}_j$ and $(s_3, s_2) \in \mathcal{R}(n)$.

Then, like for the case $n = 1$, we get $PM_A^*(s_1, \mathcal{H}) = PM_A^*(s_3, \mathcal{H})$. By the induction hypothesis, we get $PM_A^*(s_3, \mathcal{H}) = PM_A^*(s_2, \mathcal{H})$. Thus, $\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}\ \forall A \in I\!N_{fin}^{\mathcal{L}} \setminus \{\emptyset\}$

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

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

A.3 Proof of Proposition 5.3

It is sufficient to prove the statement of the proposition for $\star = s$, since $\star = i$ is a particular case of the previous one with one-element multisets of multiactions and interleaving transition relation.

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

Let $(A_1 \cdots A_n, \mathcal{Q}) \in StepProbTraces(G)$. Taking into account the notes above and $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$, we have $\forall \mathcal{H}_1, \ldots, \mathcal{H}_n \in (DR(G) \cup DR(G'))/_{\mathcal{R}}\ [G]_\approx \stackrel{A_1}{\twoheadrightarrow}_{\mathcal{P}_1} \mathcal{H}_1 \stackrel{A_2}{\twoheadrightarrow}_{\mathcal{P}_2} \cdots \stackrel{A_n}{\twoheadrightarrow}_{\mathcal{P}_n} \mathcal{H}_n \Leftrightarrow [G']_\approx \stackrel{A_1}{\twoheadrightarrow}_{\mathcal{P}_1} \mathcal{H}_1 \stackrel{A_2}{\twoheadrightarrow}_{\mathcal{P}_2} \cdots \stackrel{A_n}{\twoheadrightarrow}_{\mathcal{P}_n} \mathcal{H}_n$.

We now intend to prove that the sum of probabilities of all the paths starting in $[G]_\approx$ and going through the states from $\mathcal{H}_1, \ldots, \mathcal{H}_n$ is equal to the product of $\mathcal{P}_1, \ldots, \mathcal{P}_n$, which is essentially the probability of the "composite" path going through the equivalence classes $\mathcal{H}_1, \ldots, \mathcal{H}_n$ in $TS^*(G)$:

$$\sum_{\{\Gamma_1, \ldots, \Gamma_n \mid [G]_\approx \stackrel{\Gamma_1}{\twoheadrightarrow} \cdots \stackrel{\Gamma_n}{\twoheadrightarrow} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^n PT^*(\Gamma_i, s_{i-1}) = \prod_{i=1}^n PM_{A_i}^*(\mathcal{H}_{i-1}, \mathcal{H}_i).$$

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

- $n = 1$

  $\sum_{\{\Gamma_1 \mid [G]_\approx \stackrel{\Gamma_1}{\twoheadrightarrow} s_1,\ \mathcal{L}(\Gamma_1) = A_1,\ s_1 \in \mathcal{H}_1\}} PT^*(\Gamma_1, [G]_\approx) = PM_{A_1}^*([G]_\approx, \mathcal{H}_1) = PM_{A_1}^*(\mathcal{H}_0, \mathcal{H}_1)$.

- $n \to n + 1$

  $\sum_{\{\Gamma_1, \ldots, \Gamma_n, \Gamma_{n+1} \mid [G]_\approx \stackrel{\Gamma_1}{\twoheadrightarrow} \cdots \stackrel{\Gamma_n}{\twoheadrightarrow} s_n \stackrel{\Gamma_{n+1}}{\twoheadrightarrow} s_{n+1},\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n+1)\}} \prod_{i=1}^{n+1} PT^*(\Gamma_i, s_{i-1}) =$
  $\sum_{\{\Gamma_1, \ldots, \Gamma_n \mid [G]_\approx \stackrel{\Gamma_1}{\twoheadrightarrow} \cdots \stackrel{\Gamma_n}{\twoheadrightarrow} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \sum_{\{\Gamma_{n+1} \mid s_n \stackrel{\Gamma_{n+1}}{\twoheadrightarrow} s_{n+1},\ \mathcal{L}(\Gamma_{n+1}) = A_{n+1},\ s_n \in \mathcal{H}_n,\ s_{n+1} \in \mathcal{H}_{n+1}\}}$
  $\prod_{i=1}^n PT^*(\Gamma_i, s_{i-1}) PT^*(\Gamma_{n+1}, s_n) =$
  $\sum_{\{\Gamma_1, \ldots, \Gamma_n \mid [G]_\approx \stackrel{\Gamma_1}{\twoheadrightarrow} \cdots \stackrel{\Gamma_n}{\twoheadrightarrow} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}}$
  $\left[\prod_{i=1}^n PT^*(\Gamma_i, s_{i-1}) \sum_{\{\Gamma_{n+1} \mid s_n \stackrel{\Gamma_{n+1}}{\twoheadrightarrow} s_{n+1},\ \mathcal{L}(\Gamma_{n+1}) = A_{n+1},\ s_n \in \mathcal{H}_n,\ s_{n+1} \in \mathcal{H}_{n+1}\}} PT^*(\Gamma_{n+1}, s_n)\right] =$
  $\sum_{\{\Gamma_1, \ldots, \Gamma_n \mid [G]_\approx \stackrel{\Gamma_1}{\twoheadrightarrow} \cdots \stackrel{\Gamma_n}{\twoheadrightarrow} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^n PT^*(\Gamma_i, s_{i-1}) PM_{A_{n+1}}^*(s_n, \mathcal{H}_{n+1}) =$
  $\sum_{\{\Gamma_1, \ldots, \Gamma_n \mid [G]_\approx \stackrel{\Gamma_1}{\twoheadrightarrow} \cdots \stackrel{\Gamma_n}{\twoheadrightarrow} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^n PT^*(\Gamma_i, s_{i-1}) PM_{A_{n+1}}^*(\mathcal{H}_n, \mathcal{H}_{n+1}) =$
  $PM_{A_{n+1}}^*(\mathcal{H}_n, \mathcal{H}_{n+1}) \sum_{\{\Gamma_1, \ldots, \Gamma_n \mid [G]_\approx \stackrel{\Gamma_1}{\twoheadrightarrow} \cdots \stackrel{\Gamma_n}{\twoheadrightarrow} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^n PT^*(\Gamma_i, s_{i-1}) =$
  $PM_{A_{n+1}}^*(\mathcal{H}_n, \mathcal{H}_{n+1}) \prod_{i=1}^n PM_{A_i}^*(\mathcal{H}_{i-1}, \mathcal{H}_i) = \prod_{i=1}^{n+1} PM_{A_i}^*(\mathcal{H}_{i-1}, \mathcal{H}_i)$.

Note that the equality we have just proved can also be applied to G'.

We now only need to see that the summation over *all multisets of activities* is the same as the summation over *all equivalence classes*: $\mathcal{Q} = \sum_{\{\Gamma_1, \ldots, \Gamma_n \mid [G]_\approx \stackrel{\Gamma_1}{\twoheadrightarrow} \cdots \stackrel{\Gamma_n}{\twoheadrightarrow} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ (1 \le i \le n)\}} \prod_{i=1}^n PT^*(\Gamma_i, s_{i-1}) =$
$\sum_{\mathcal{H}_1, \ldots, \mathcal{H}_n} \sum_{\{\Gamma_1, \ldots, \Gamma_n \mid [G]_\approx \stackrel{\Gamma_1}{\twoheadrightarrow} \cdots \stackrel{\Gamma_n}{\twoheadrightarrow} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^n PT^*(\Gamma_i, s_{i-1}) =$
$\sum_{\mathcal{H}_1, \ldots, \mathcal{H}_n} \prod_{i=1}^n PM_{A_i}^*(\mathcal{H}_{i-1}, \mathcal{H}_i) =$
$\sum_{\mathcal{H}_1, \ldots, \mathcal{H}_n} \sum_{\{\Gamma'_1, \ldots, \Gamma'_n \mid [G']_\approx \stackrel{\Gamma'_1}{\twoheadrightarrow} \cdots \stackrel{\Gamma'_n}{\twoheadrightarrow} s'_n,\ \mathcal{L}(\Gamma'_i) = A_i,\ s'_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^n PT^*(\Gamma'_i, s'_{i-1}) =$
$\sum_{\{\Gamma'_1, \ldots, \Gamma'_n \mid [G']_\approx \stackrel{\Gamma'_1}{\twoheadrightarrow} \cdots \stackrel{\Gamma'_n}{\twoheadrightarrow} s'_n,\ \mathcal{L}(\Gamma'_i) = A_i,\ (1 \le i \le n)\}} \prod_{i=1}^n PT^*(\Gamma'_i, s'_{i-1})$.

Hence, $(A_1 \cdots A_n, \mathcal{Q}) \in StepProbTraces(G')$, and we have $StepProbTraces(G) \subseteq StepProbTraces(G')$. The reverse inclusion is proved by symmetry. □

A.4 Proof of Proposition 8.1

The proof is an extension of results from [41] to the process algebra framework and discrete time case.

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

- $k = 0$

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

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

- $k \to k + 1$

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

  By induction hypothesis, $\sum_{s\in\mathcal{H}\cap DR(G)} \psi^*[k](s) = \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'^*[k](s')$. Further,
  $\sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} \psi^*[k+1](\tilde{s}) = \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} \sum_{s\in DR(G)} \psi^*[k](s)PM^*(s,\tilde{s}) =$
  $\sum_{s\in DR(G)} \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} \psi^*[k](s)PM^*(s,\tilde{s}) = \sum_{s\in DR(G)} \psi^*[k](s) \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} PM^*(s,\tilde{s}) =$
  $\sum_{\mathcal{H}} \sum_{s\in\mathcal{H}\cap DR(G)} \psi^*[k](s) \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} PM^*(s,\tilde{s}) =$
  $\sum_{\mathcal{H}} \sum_{s\in\mathcal{H}\cap DR(G)} \psi^*[k](s) \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} \sum_{\{\Gamma\mid s\stackrel{\Gamma}{\twoheadrightarrow}\tilde{s}\}} PT^*(\Gamma, s) =$
  $\sum_{\mathcal{H}} \sum_{s\in\mathcal{H}\cap DR(G)} \psi^*[k](s) \sum_{\{\Gamma\mid\exists\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)\ s\stackrel{\Gamma}{\twoheadrightarrow}\tilde{s}\}} PT^*(\Gamma, s) =$
  $\sum_{\mathcal{H}} \sum_{s\in\mathcal{H}\cap DR(G)} \psi^*[k](s)PM^*(s,\widetilde{\mathcal{H}}) = \sum_{\mathcal{H}} \sum_{s\in\mathcal{H}\cap DR(G)} \psi^*[k](s)PM^*(\mathcal{H},\widetilde{\mathcal{H}}) =$
  $\sum_{\mathcal{H}} PM^*(\mathcal{H},\widetilde{\mathcal{H}}) \sum_{s\in\mathcal{H}\cap DR(G)} \psi^*[k](s) = \sum_{\mathcal{H}} PM^*(\mathcal{H},\widetilde{\mathcal{H}}) \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'^*[k](s') =$
  $\sum_{\mathcal{H}} \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'^*[k](s')PM^*(\mathcal{H},\widetilde{\mathcal{H}}) = \sum_{\mathcal{H}} \sum_{s'\in\mathcal{H}'\cap DR(G')} \psi'^*[k](s')PM^*(s',\widetilde{\mathcal{H}}) =$
  $\sum_{\mathcal{H}} \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'^*[k](s') \sum_{\{\Gamma\mid\exists\tilde{s}'\in\widetilde{\mathcal{H}}\cap DR(G')\ s'\stackrel{\Gamma}{\twoheadrightarrow}\tilde{s}'\}} PT^*(\Gamma, s') =$
  $\sum_{\mathcal{H}} \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'^*[k](s') \sum_{\tilde{s}'\in\widetilde{\mathcal{H}}\cap DR(G')} \sum_{\{\Gamma\mid\exists\tilde{s}'\ s'\stackrel{\Gamma}{\twoheadrightarrow}\tilde{s}'\}} PT^*(\Gamma, s') =$
  $\sum_{\mathcal{H}} \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'^*[k](s') \sum_{\tilde{s}'\in\widetilde{\mathcal{H}}\cap DR(G')} PM^*(s',\tilde{s}') =$
  $\sum_{s'\in DR(G')} \psi'^*[k](s') \sum_{\tilde{s}'\in\widetilde{\mathcal{H}}\cap DR(G')} PM^*(s',\tilde{s}') = \sum_{s'\in DR(G')} \sum_{\tilde{s}'\in\widetilde{\mathcal{H}}\cap DR(G')} \psi'^*[k](s')PM^*(s',\tilde{s}') =$
  $\sum_{\tilde{s}'\in\widetilde{\mathcal{H}}\cap DR(G')} \sum_{s'\in DR(G')} \psi'^*[k](s')PM^*(s',\tilde{s}') = \sum_{\tilde{s}'\in\widetilde{\mathcal{H}}\cap DR(G')} \psi'^*[k+1](\tilde{s}')$. ☐

A.5 Proof of Theorem 8.2

The main idea of the proof is similar to that from [36, 37] but in the algebraic setting.

Let $\mathcal{H} \in (DR(G) \cup DR(G'))/_\mathcal{R}$ and $s, \bar{s} \in \mathcal{H}$. We have $\forall\widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_\mathcal{R}\ \forall A \in I\!N^{\mathcal{L}}_{fin} \setminus \{\emptyset\}$ $s \stackrel{A}{\twoheadrightarrow}_\mathcal{P} \widetilde{\mathcal{H}} \Leftrightarrow \bar{s} \stackrel{A}{\twoheadrightarrow}_\mathcal{P} \widetilde{\mathcal{H}}$. The previous equality is valid for all $s, \bar{s} \in \mathcal{H}$, hence, we can rewrite it as $\mathcal{H} \stackrel{A}{\twoheadrightarrow}_\mathcal{P} \widetilde{\mathcal{H}}$ and denote $PM^*_A(\mathcal{H},\widetilde{\mathcal{H}}) = PM^*_A(s,\widetilde{\mathcal{H}}) = PM^*_A(\bar{s},\widetilde{\mathcal{H}})$. Note that transitions from the states of $DR(G)$ always lead to those from the same set, hence, $\forall s \in DR(G)\ PM^*_A(s,\widetilde{\mathcal{H}}) = PM^*_A(s,\widetilde{\mathcal{H}} \cap DR(G))$. The same is true for $DR(G')$.

Let $\Sigma = A_1 \cdots A_n$ be a step trace of G and G'. We have $\exists\mathcal{H}_0, \ldots, \mathcal{H}_n \in (DR(G) \cup DR(G'))/_\mathcal{R}$ $\mathcal{H}_0 \stackrel{A_1}{\twoheadrightarrow}_{\mathcal{P}_1} \mathcal{H}_1 \stackrel{A_2}{\twoheadrightarrow}_{\mathcal{P}_2} \cdots \stackrel{A_n}{\twoheadrightarrow}_{\mathcal{P}_n} \mathcal{H}_n$. We now intend to prove that the sum of probabilities of all the paths starting in every $s_0 \in \mathcal{H}_0$ and going through the states from $\mathcal{H}_1, \ldots, \mathcal{H}_n$ is equal to the product of $\mathcal{P}_1, \ldots, \mathcal{P}_n$:

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

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

- $n=1$

  $\sum_{\{\Gamma_1|s_0\stackrel{\Gamma_1}{\twoheadrightarrow}s_1,\ \mathcal{L}(\Gamma_1)=A_1,\ s_1\in\mathcal{H}_1\}} PT^*(\Gamma_1,s_0)=PM^*_{A_1}(s_0,\mathcal{H}_1)=PM^*_{A_1}(\mathcal{H}_0,\mathcal{H}_1).$

- $n\to n+1$

  $\sum_{\{\Gamma_1,\dots,\Gamma_n,\Gamma_{n+1}|s_0\stackrel{\Gamma_1}{\twoheadrightarrow}\cdots\stackrel{\Gamma_n}{\twoheadrightarrow}s_n\stackrel{\Gamma_{n+1}}{\twoheadrightarrow}s_{n+1},\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\le i\le n+1)\}} \prod_{i=1}^{n+1} PT^*(\Gamma_i,s_{i-1})=$
  $\sum_{\{\Gamma_1,\dots,\Gamma_n|s_0\stackrel{\Gamma_1}{\twoheadrightarrow}\cdots\stackrel{\Gamma_n}{\twoheadrightarrow}s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\le i\le n)\}} \sum_{\{\Gamma_{n+1}|s_n\stackrel{\Gamma_{n+1}}{\twoheadrightarrow}s_{n+1},\ \mathcal{L}(\Gamma_{n+1})=A_{n+1},\ s_n\in\mathcal{H}_n,\ s_{n+1}\in\mathcal{H}_{n+1}\}}$
  $\prod_{i=1}^{n} PT^*(\Gamma_i,s_{i-1})PT^*(\Gamma_{n+1},s_n)=$
  $\sum_{\{\Gamma_1,\dots,\Gamma_n|s_0\stackrel{\Gamma_1}{\twoheadrightarrow}\cdots\stackrel{\Gamma_n}{\twoheadrightarrow}s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\le i\le n)\}}$
  $\left[\prod_{i=1}^{n} PT^*(\Gamma_i,s_{i-1})\sum_{\{\Gamma_{n+1}|s_n\stackrel{\Gamma_{n+1}}{\twoheadrightarrow}s_{n+1},\ \mathcal{L}(\Gamma_{n+1})=A_{n+1},\ s_n\in\mathcal{H}_n,\ s_{n+1}\in\mathcal{H}_{n+1}\}} PT^*(\Gamma_{n+1},s_n)\right]=$
  $\sum_{\{\Gamma_1,\dots,\Gamma_n|s_0\stackrel{\Gamma_1}{\twoheadrightarrow}\cdots\stackrel{\Gamma_n}{\twoheadrightarrow}s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\le i\le n)\}} \prod_{i=1}^{n} PT^*(\Gamma_i,s_{i-1})PM^*_{A_{n+1}}(s_n,\mathcal{H}_{n+1})=$
  $\sum_{\{\Gamma_1,\dots,\Gamma_n|s_0\stackrel{\Gamma_1}{\twoheadrightarrow}\cdots\stackrel{\Gamma_n}{\twoheadrightarrow}s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\le i\le n)\}} \prod_{i=1}^{n} PT^*(\Gamma_i,s_{i-1})PM^*_{A_{n+1}}(\mathcal{H}_n,\mathcal{H}_{n+1})=$
  $PM^*_{A_{n+1}}(\mathcal{H}_n,\mathcal{H}_{n+1})\sum_{\{\Gamma_1,\dots,\Gamma_n|s_0\stackrel{\Gamma_1}{\twoheadrightarrow}\cdots\stackrel{\Gamma_n}{\twoheadrightarrow}s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\le i\le n)\}} \prod_{i=1}^{n} PT^*(\Gamma_i,s_{i-1})=$
  $PM^*_{A_{n+1}}(\mathcal{H}_n,\mathcal{H}_{n+1})\prod_{i=1}^{n} PM^*_{A_i}(\mathcal{H}_{i-1},\mathcal{H}_i)=\prod_{i=1}^{n+1} PM^*_{A_i}(\mathcal{H}_{i-1},\mathcal{H}_i).$

Let $s_0,\bar{s}_0\in\mathcal{H}_0$. We have
$PT^*(A_1\cdots A_n,s_0)=\sum_{\{\Gamma_1,\dots,\Gamma_n|s_0\stackrel{\Gamma_1}{\twoheadrightarrow}\cdots\stackrel{\Gamma_n}{\twoheadrightarrow}s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ (1\le i\le n)\}} \prod_{i=1}^{n} PT^*(\Gamma_i,s_{i-1})=$
$\sum_{\mathcal{H}_1,\dots,\mathcal{H}_n}\sum_{\{\Gamma_1,\dots,\Gamma_n|s_0\stackrel{\Gamma_1}{\twoheadrightarrow}\cdots\stackrel{\Gamma_n}{\twoheadrightarrow}s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i\in\mathcal{H}_i\ (1\le i\le n)\}} \prod_{i=1}^{n} PT^*(\Gamma_i,s_{i-1})=$
$\sum_{\mathcal{H}_1,\dots,\mathcal{H}_n}\prod_{i=1}^{n} PM^*_{A_i}(\mathcal{H}_{i-1},\mathcal{H}_i)=$
$\sum_{\mathcal{H}_1,\dots,\mathcal{H}_n}\sum_{\{\overline{\Gamma}_1,\dots,\overline{\Gamma}_n|\bar{s}_0\stackrel{\overline{\Gamma}_1}{\twoheadrightarrow}\cdots\stackrel{\overline{\Gamma}_n}{\twoheadrightarrow}\bar{s}_n,\ \mathcal{L}(\overline{\Gamma}_i)=A_i,\ \bar{s}_i\in\mathcal{H}_i\ (1\le i\le n)\}} \prod_{i=1}^{n} PT^*(\overline{\Gamma}_i,\bar{s}_{i-1})=$
$\sum_{\{\overline{\Gamma}_1,\dots,\overline{\Gamma}_n|\bar{s}_0\stackrel{\overline{\Gamma}_1}{\twoheadrightarrow}\cdots\stackrel{\overline{\Gamma}_n}{\twoheadrightarrow}\bar{s}_n,\ \mathcal{L}(\overline{\Gamma}_i)=A_i,\ (1\le i\le n)\}} \prod_{i=1}^{n} PT^*(\overline{\Gamma}_i,\bar{s}_{i-1})=PT^*(A_1\cdots A_n,\bar{s}_0).$
Since we have the previous equality for all $s_0,\bar{s}_0\in\mathcal{H}_0$, we can denote $PT^*(A_1\cdots A_n,\mathcal{H}_0)=PT^*(A_1\cdots A_n,s_0)=PT^*(A_1\cdots A_n,\bar{s}_0)$.

By Proposition 8.1, $\sum_{s\in\mathcal{H}\cap DR(G)}\psi^*(s)=\sum_{s'\in\mathcal{H}\cap DR(G')}\psi'^*(s')$. We now can complete the proof:
$\sum_{s\in\mathcal{H}\cap DR(G)}\psi^*(s)PT^*(\Sigma,s)=\sum_{s\in\mathcal{H}\cap DR(G)}\psi^*(s)PT^*(\Sigma,\mathcal{H})=PT^*(\Sigma,\mathcal{H})\sum_{s\in\mathcal{H}\cap DR(G)}\psi^*(s)=$
$PT^*(\Sigma,\mathcal{H})\sum_{s'\in\mathcal{H}\cap DR(G')}\psi'^*(s')=\sum_{s'\in\mathcal{H}\cap DR(G')}\psi'^*(s')PT^*(\Sigma,\mathcal{H})=\sum_{s'\in\mathcal{H}\cap DR(G')}\psi'^*(s')PT^*(\Sigma,s')$. □

A.6 Proof of Proposition 8.2

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

We now prove the proposition statement for the sojourn time averages. Let $\mathcal{H}\in(DR(G)\cup DR(G'))/_\mathcal{R}$. We have $\mathcal{H}\cap DR(G)\in DR(G)/_\mathcal{R}$ and $\mathcal{H}\cap DR(G')\in DR(G')/_\mathcal{R}$. By definition of the average sojourn time in an equivalence class of states, we get $SJ_{\mathcal{R}\cap(DR(G))^2}(\mathcal{H}\cap DR(G))=\frac{1}{1-PM(\mathcal{H}\cap DR(G),\mathcal{H}\cap DR(G))}=\frac{1}{1-PM(\mathcal{H},\mathcal{H})}=\frac{1}{1-PM(\mathcal{H}\cap DR(G'),\mathcal{H}\cap DR(G'))}=SJ_{\mathcal{R}\cap(DR(G'))^2}(\mathcal{H}\cap DR(G'))$.

The proposition statement for the sojourn time variances is proved similarly to that for the averages. □