CARL VON OSSIETZKY universität OLDENBURG

BERICHTE

AUS DEM DEPARTMENT FÜR INFORMATIK
der Fakultät II - Informatik, Wirtschafts- und Rechtswissenschaften

Herausgeber: Die Professorinnen und Professoren
des Departments für Informatik

Discrete time stochastic Petri box calculus

Dr. Igor V. Tarasyuk

Bericht

Nummer 03/05 – November 2005

ISSN 0946-2910

BERICHTE

AUS DEM DEPARTMENT FÜR INFORMATIK

der Fakultät II - Informatik, Wirtschafts- und Rechtswissenschaften

Herausgeber: Die Professorinnen und Professoren
des Departments für Informatik

Discrete time stochastic Petri box calculus

Dr. Igor Tarasyuk

Bericht

Nummer 03/05 – November 2005

Discrete time stochastic Petri box calculus *

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

Last decade, a number of stochastic enrichments of process algebras was constructed to facilitate the specification of stochastic processes based on the the well-developed framework of algebraic calculi. In [56], a continuous time stochastic extension of finite PBC was proposed called $sPBC$. Algebra $sPBC$ has interleaving semantics due to the properties of continuous time distributions. At the same time, PBC has step semantics, and it could be natural to propose its concurrent stochastic enrichment. We construct a discrete time stochastic extension $dtsPBC$ of finite PBC. Step operational semantics is defined in terms of labeled transition systems based on action and inaction rules. Denotational semantics is defined in terms of a subclass of labeled DTSPNs (LDTSPNs) called discrete time stochastic Petri boxes (dts-boxes). An accordance of both the semantics is demonstrated. In addition, we define a variety of probabilistic equivalences that allow one to identify stochastic processes with similar behaviour that are differentiated by too strict notion of the semantic equivalence. The interrelations of all the introduced equivalences are investigated. Some of the relations could be later considered as candidates for the role of congruence.

Keywords: Stochastic Petri nets, stochastic process algebras, Petri box calculus, discrete time, transition systems, operational semantics, dts-boxes, denotational semantics, empty loops, probabilistic equivalences.

1 Introduction

Stochastic Petri nets (SPNs) are a well-known model for quantitative analysis of discrete dynamic event systems proposed in [45, 46, 26]. 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 analysed 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 and discrete 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 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 step semantics. In this semantics, parallel computations are modeled by sequences of concurrent occurrences (steps) of their components. In [18, 19], a labeling for transitions of CTSPNs with action names was proposed. 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 quantitative and qualitative analysis.

Algebraic calculi hold 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. A verification of the properties is accomplished at a syntactic level by means of well-developed systems of equivalences, axioms and inference rules. One of the first PAs was CCS (Calculus of Communicating Systems) [44]. Process algebras has been acknowledged to be very suitable formalism to operate with real

*This work was supported by Deutscher Akademischer Austauschdienst (DAAD), grant A/05/05334

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 that can happen (qualitative features) as usual process algebras, but they associate some quantitative parameters with actions (quantitative characteristics). The papers [31, 17, 29, 23, 62, 11] propose a variety of SPAs. 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, since it allows to combine advantages of the both models. For this, 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.

As a rule, stochastic process calculi proposed in the literature are interleaving. As a semantic area, the interleaving formalism of transition systems is used. For example, an extension of CCS with probabilities and time called $TPCCS$ was defined in [28]. An enrichment of BPA with probabilistic choice, $prBPA$, as well as an extension of $prBPA$ with parallel composition operator named ACP_π^+ have been proposed in [1]. A standard way for probabilistic extension of process algebras into the calculi of probabilistic transition systems was described in [32]. The most famous SPAs proposed so far are $PEPA$ [29], $TIPP$ [31] and $EMPA$ [10]. It is worth to mention the stochastic process calculus PPA constructed in [59, 61] as well. Therefore, an investigation of a stochastic extension for more expressive and powerful algebraic calculi is very important. At present, the development of step or "true concurrent" (such that parallelism is considered as a causal independence) SPAs is in the very beginning. One can mention a concurrent SPA of finite processes $StAFP_0$ with step semantics proposed in [16]. At the same time, there still exists no algebra of infinite concurrent stochastic processes.

Petri box calculus (PBC) is a flexible and expressive process algebra based on calculi CCS [44] and AFP_0 [36]. PBC was proposed more than 10 years ago [3], and it was well explored since that time [2, 14, 20, 35, 37, 12, 13, 21, 22, 24, 30, 4, 5, 33, 6, 7, 8, 9]. It was intended to become a tool for description of a Petri net structure and relationships between nets. 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 only, as in CCS, but from multisets of actions called multiactions (*basic formulas*) as well. 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 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, algebra PBC has an additional iteration construction to specify infiniteness in the cases when finite Petri nets can be used as the semantic interpretation. For PBC, denotational semantics in terms of a subclass of Petri nets equipped with interface and considered up to isomorphism was proposed. This subclass is called Petri boxes. Calculus PBC has step operational semantics in terms of labeled transition systems based on structural operational semantics (SOS) rules. Pomset operational semantics of PBC was defined in [37] 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. Last years, several extensions of PBC were presented.

A time extension of PBC called time Petri box calculus ($tPBC$) was proposed in [38]. In $tPBC$, timing information is added by combining instantaneous multiactions and time delays. Denotational semantics was defined in terms of a subclass of labeled time Petri nets (tPNs) called time Petri boxes (ct-boxes). $tPBC$ has interleaving time operational semantics in terms of labeled transition systems. Another time enrichment of PBC called Timed Petri box calculus ($TPBC$) was defined in [42, 43]. 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 the calculus appeared to be more complicated than $tPBC$. Denotational semantics was defined in terms of a subclass of labeled Timed Petri nets (TPNs) called Timed Petri boxes (T-boxes). Algebra $tPBC$ has 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 [60]. In $atPBC$, multiactions are associated with time delay intervals. Denotational semantics was defined on a subclass of arc time Petri nets (atPNs) called arc time Petri boxes (at-boxes). $atPBC$ possesses a step operational semantics in terms of labeled transition systems.

A stochastic extension of PBC called stochastic Petri box calculus ($sPBC$) was proposed in [56, 57, 58, 47, 52, 53, 54, 41]. In $sPBC$, multiactions have stochastic durations that follow negative exponential distribution. Each multiaction is instantaneous and equipped with a rate that is a parameter of the corresponding exponential distribution. The execution of a 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 or recursion or iteration operations. Denotational semantics was defined in terms of a subclass of labeled continuous time stochastic Petri nets (CTSPNs) called stochastic Petri boxes (s-boxes). Calculus $sPBC$ has interleaving operational semantics in terms of labeled transition systems. Note that we have interleaving behaviour here because of the fact that a simultaneous firing of any two transitions has zero probability in accordance to the properties of continuous time distributions. Current research in this branch has an aim to extend the specification abilities of $sPBC$ and to define an appropriate congruence relation over algebraic formulas. Recent results on constructing iteration for $sPBC$ were reported in [49, 50]. In the papers [48, 51], a number of new equivalence relations were proposed for regular terms of $sPBC$ to choose later a suitable candidate for a congruence. In [55] special multiactions with zero time delay were added to $sPBC$. Denotational semantics of such a $sPBC$ extension was defined via a subclass of labeled generalized SPNs (GSPNs). The subclass is called generalized stochastic Petri boxes (gs-boxes).

An ambient extension of PBC called Ambient Petri box calculus ($APBC$) was proposed in [25]. Ambient calculus is used to model behaviour of mobile systems. Ambient is an named environment delimited by a boundary. The ambients can be moved to a new location thus modeling mobility. 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, just concurrency and sequence were transferred into $APBC$ from the set of PBC operations in [25]. This reduced algebra was called Simple Ambient Petri box calculus ($SAPBC$). 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. Calculus $SAPBC$ has step operational semantics in terms of labeled transition systems.

Nevertheless, there is still no stochastic extension of PBC with step semantics. It could be done with the use of labeled DTSPNs as a semantic area, since discrete time models allow for concurrent action occurrences. The enrichment based of DTSPNs would be natural because PBC has step denotational and 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 were well explored in the literature. The most famous 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 untimed 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 the probabilities summation. 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 [11], a notion of interleaving stochastic 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. 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 relates 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 helpful 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 [15], a new net class was proposed called discrete time weighted SPNs (DTWSPNs) that is a modification of DTSPNs by transition labeling and weights. Transitions of DTWSPNs are labeled by actions that represent elementary activities and can be visible or invisible to an external observer. For this net class, 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 stationary behaviour of the DTWSPNs was demonstrated. In [65, 67], 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 DTWSPNs are (interleaving or step) probabilistic τ-bisimulation equivalent if they satisfy the same formulas of the corresponding probabilistic modal logic. Thus, instead of comparing DTWSPNs 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 proposed in [39] on probabilistic transition systems with visible actions. In [16, 67], a stochastic algebra of finite nondeterministic processes $StAFP_0$ was proposed with semantics in terms of a subclasses of DTWSPNs called stochastic acyclic nets (SANs). The calculus defined is a stochastic extension of algebra AFP_0 introduced in [34]. Calculus $StAFP_0$ 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 concurrent" 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 [66], 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 this paper, we propose a discrete time stochastic extension of finite PBC called $dtsPBC$. The work consists of the following stages. First, we present syntax of $dtsPBC$. Each multiaction of the initial calculus PBC is associated with a conditional probability. Such a pair is called stochastic multiaction or activity. Second, we propose semantics of $dtsPBC$. Step operational semantics is constructed in terms of labeled transition systems based on action and inaction rules. The complexity here is a careful elaboration of step probabilities for formulas with parallelism and synchronization as well as the conflict resolving mechanism related to the probabilistic choice. Denotational semantics is defined in terms of a subclass of labeled DTSPNs (LDTSPNs) called discrete time stochastic Petri boxes (dts-boxes). An accordance of operational and denotational semantics is proved. At last, we define a number of probabilistic equivalences in the algebraic setting based of 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 strict in many cases. The interrelations diagram of all the introduced equivalences is built. Moreover, the proposed equivalences could be used to construct later a congruence relation based on one of them. In the best case, a complete and correct finite axiomatization of the congruence could be constructed. The hard task here would be to find a congruence that is not too distinctive, i.e., it should differentiate formulas with really different behaviour only in accordance to our needs. Moreover, the relation is to be axiomatizable and easy to check.

The paper is organized as follows. In the next Section 2 a syntax of calculus $dtsPBC$ is presented. Then, in Section 3 we construct operational semantics of the algebra in terms of labeled transition systems. In Section 4 we propose denotational semantics based on a subclass of LDTSPNs. Section 5 is devoted to the construction and the interrelations of probabilistic algebraic equivalences based on transition systems without empty loops. The concluding Section 6 summarizes the results obtained and outlines research perspectives in this area.

2 Syntax

Petri box calculus PBC was proposed in [3]. Its formulas specify Petri boxes (PBs), a special class of labeled Petri nets. In this section we propose a syntax of discrete time stochastic extension of finite PBC called *discrete time stochastic Petri box calculus $dtsPBC$* with semantics in terms of discrete time stochastic Petri boxes (dtsPBs), a special class of LDTSPNs.

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 finite number of elements only.*

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

Let $Act = \{a, b, \ldots\}$ be the set of *elementary actions*. Then $\widehat{Act} = \{\hat{a}, \hat{b}, \ldots\}$ be the set of *conjunctive actions (conjugates)* such that $a \ne \hat{a}$ and $\hat{\hat{a}} = a$. Let $\mathcal{A} = Act \cup \widehat{Act}$ be the set of *all actions*, and $\mathcal{L} = I\!N_f^{\mathcal{A}}$ be the set of *all multiactions*. Note that $\emptyset \in \mathcal{L}$, this corresponds to an internal activity, i.e., the execution of a multiaction than 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 multiaction α. Let $\mathcal{SL}$ be the set of *all activities*. Let us note that the same multiaction $\alpha \in \mathcal{L}$ may have different probabilities in the same specification. The *alphabet* of $(\alpha, \rho) \in \mathcal{SL}$ is defined as $\mathcal{A}(\alpha, \rho) = \mathcal{A}(\alpha)$. For $(\alpha, \rho) \in \mathcal{SL}$, we define its *multiaction part* as $\mathcal{L}(\alpha, \rho) = \alpha$ and its *probability part* as $\Omega(\alpha, \rho) = \rho$.

Activities are combined into formulas by the following operations: *sequential execution* ;, *choice* [], *parallelism* $\|$, *relabeling* $[f]$, *synchronization sy* and *restriction rs*.

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

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

Static expressions specify structure of a system. As we shall see, they correspond to unmarked SPNs.

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

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

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

Dynamic expressions specify current state of a system. As we shall see, they correspond to marked SPNs.

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

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

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

3 Operational semantics

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

3.1 Inaction rules

First, we define inaction rules without preconditions. Let $E, F \in StatExpr,\ G \in DynExpr$ and $a \in Act$.

$$\begin{array}{lllll}
\overline{E;F} \xrightarrow{\emptyset} \overline{E};F & \underline{E};F \xrightarrow{\emptyset} E;\overline{F} & E;\underline{F} \xrightarrow{\emptyset} \underline{E;F} & \overline{E[]F} \xrightarrow{\emptyset} \overline{E}[]F & \overline{E[]F} \xrightarrow{\emptyset} E[]\overline{F} \\
\underline{E}[]F \xrightarrow{\emptyset} \underline{E[]F} & E[]\underline{F} \xrightarrow{\emptyset} \underline{E[]F} & \overline{E\|F} \xrightarrow{\emptyset} \overline{E}\|\overline{F} & \underline{E}\|\underline{F} \xrightarrow{\emptyset} \underline{E\|F} & \overline{E[f]} \xrightarrow{\emptyset} \overline{E}[f] \\
\underline{E}[f] \xrightarrow{\emptyset} \underline{E[f]} & \overline{E \ \mathsf{rs}\ a} \xrightarrow{\emptyset} \overline{E} \ \mathsf{rs}\ a & \underline{E} \ \mathsf{rs}\ a \xrightarrow{\emptyset} \underline{E \ \mathsf{rs}\ a} & \overline{E \ \mathsf{sy}\ a} \xrightarrow{\emptyset} \overline{E} \ \mathsf{sy}\ a & \underline{E} \ \mathsf{sy}\ a \xrightarrow{\emptyset} \underline{E \ \mathsf{sy}\ a} \\
G \xrightarrow{\emptyset} G & & & &
\end{array}$$

Note that the rule $G \xrightarrow{\emptyset} G$ is intentionally included in the set of rules above. It reflects a non-zero probability to stay in a state at the next time moment that is an essential feature of discrete time stochastic processes.

Second, we propose inaction rules with preconditions. Let $E \in StatExpr,\ G, H, \widetilde{G}, \widetilde{H} \in DynExpr$ and $a \in Act$.

$$\frac{G \xrightarrow{\emptyset} \widetilde{G},\ \circ \in \{;, []\}}{G \circ E \xrightarrow{\emptyset} \widetilde{G} \circ E} \quad \frac{G \xrightarrow{\emptyset} \widetilde{G},\ \circ \in \{;, []\}}{E \circ G \xrightarrow{\emptyset} E \circ \widetilde{G}} \quad \frac{G \xrightarrow{\emptyset} \widetilde{G}}{G\|H \xrightarrow{\emptyset} \widetilde{G}\|H} \quad \frac{H \xrightarrow{\emptyset} \widetilde{H}}{G\|H \xrightarrow{\emptyset} G\|\widetilde{H}} \quad \frac{G \xrightarrow{\emptyset} \widetilde{G}}{G[f] \xrightarrow{\emptyset} \widetilde{G}[f]} \quad \frac{G \xrightarrow{\emptyset} \widetilde{G},\ \circ \in \{\mathsf{rs}, \mathsf{sy}\}}{G \circ a \xrightarrow{\emptyset} \widetilde{G} \circ a}$$

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

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

3.2 Action rules

Now we propose action rules that describe expression transformations due to the execution of multisets of activities. Let $(\alpha, \rho), (\beta, \chi) \in \mathcal{SL},\ E \in StatExpr,\ G, H \in OpDynExpr,\ \widetilde{G}, \widetilde{H} \in DynExpr$ and $a \in Act$. Moreover, let $\Gamma, \Delta \in I\!N_f^{\mathcal{SL}}$. The *alphabet* of $\Gamma \in I\!N_f^{\mathcal{SL}}$ is defined as $\mathcal{A}(\Gamma) = \cup_{(\alpha,\rho)\in\Gamma} \mathcal{A}(\alpha)$.

$$\frac{}{\overline{(\alpha,\rho)} \xrightarrow{\{(\alpha,\rho)\}} \underline{(\alpha,\rho)}} \quad \frac{G \xrightarrow{\Gamma} \widetilde{G}}{G;E \xrightarrow{\Gamma} \widetilde{G};E} \quad \frac{G \xrightarrow{\Gamma} \widetilde{G}}{E;G \xrightarrow{\Gamma} E;\widetilde{G}} \quad \frac{G \xrightarrow{\Gamma} \widetilde{G}}{G[]E \xrightarrow{\Gamma} \widetilde{G}[]E} \quad \frac{G \xrightarrow{\Gamma} \widetilde{G}}{E[]G \xrightarrow{\Gamma} E[]\widetilde{G}} \quad \frac{G \xrightarrow{\Gamma} \widetilde{G}}{G\|H \xrightarrow{\Gamma} \widetilde{G}\|H} \quad \frac{H \xrightarrow{\Gamma} \widetilde{H}}{G\|H \xrightarrow{\Gamma} G\|\widetilde{H}}$$

$$\frac{G \xrightarrow{\Gamma} \widetilde{G},\ H \xrightarrow{\Delta} \widetilde{H}}{G\|H \xrightarrow{\Gamma+\Delta} \widetilde{G}\|\widetilde{H}} \quad \frac{G \xrightarrow{\Gamma} \widetilde{G}}{G[f] \xrightarrow{f(\Gamma)} \widetilde{G}[f]} \quad \frac{G \xrightarrow{\Gamma} \widetilde{G},\ a,\hat{a} \notin \mathcal{A}(\Gamma)}{G \text{ rs } a \xrightarrow{\Gamma} \widetilde{G} \text{ rs } a} \quad \frac{G \xrightarrow{\Gamma} \widetilde{G}}{G \text{ sy } a \xrightarrow{\Gamma} \widetilde{G} \text{ sy } a}$$

$$\frac{G \text{ sy } a \xrightarrow{\Gamma+\{(\alpha,\rho)\}+\{(\beta,\chi)\}} \widetilde{G} \text{ sy } a,\ a \in \mathcal{A}(\alpha),\ \hat{a} \in \mathcal{A}(\beta)}{G \text{ sy } a \xrightarrow{\Gamma+\{(\alpha\oplus_a\beta,\rho\cdot\chi)\}} \widetilde{G} \text{ sy } a}$$

Note that in the last rule we multiply probabilities of synchronized multiactions since this corresponds to the probability of event intersection.

3.3 Transition systems

Now we define transition systems associated with dynamic expressions.

Note that expressions of *dtsPBC* can contain identical activities. To avoid technical difficulties such as those with proper calculation of state change probabilities for multiple transitions, we can always enumerate coinciding activities from left to right in the syntax of expressions. In the following, we suppose that all identical activities are enumerated. In the case new transitions are produced by synchronization, the new enumeration is added as suffix to the old one, if needed. Note that after such the enumeration the multisets of activities which change expressions in accordance to the action rules will be proper sets.

Let G be a dynamic expression. Then $[G]_\simeq = \{H \mid G \simeq H\}$ is the equivalence class of G with respect to isomorphism.

Definition 3.1 *The* derivation set *of a dynamic expression G, denoted by $DR(G)$, is the minimal set such that*

- $[G]_\simeq \in DR(G)$*;*
- *if* $[H]_\simeq \in DR(G)$ *and* $\exists\Gamma\ H \xrightarrow{\Gamma} \widetilde{H}$ *then* $[\widetilde{H}]_\simeq \in DR(G)$*.*

Let G be a dynamic expression and $[H]_\simeq \in DR(G)$.

The set of *all multisets of activities executable from H* is defined as $Exec(H) = \{\Gamma \mid \exists J \in [H]_\simeq, \widetilde{J}\ J \xrightarrow{\Gamma} \widetilde{J}\}$.

Let $\Gamma \in Exec(H)$. The *conditional* probability that the activities from Γ happen in H (the case when no activities conflicting with those from Γ can happen) is

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

Thus, $PF(\Gamma, H)$ could be interpreted as a *joint* probability of independent events. Each such an event is interpreted as executing or not executing of a particular activity from Γ. The multiplication in the definition is because it reflects the probability of event intersection.

The *normalized* probability that the activities from Γ happen in H is

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

Thus, $PT(\Gamma, H)$ is the probability that the multiset of activities Γ is executed *normalized* by the probability that *any* executable from H multiset occurs. The denominator of the fraction above is a summation since it reflects the probability of the event union. The definition of $PT(\Gamma, H)$, unlike that of $PF(\Gamma, H)$, respects the fact that some activities from a multiset belonging to $Exec(H)$ could be in conflict with those from another executable multiset and, hence, cannot be fired together.

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

The probability that the execution of *any* activities changes H by $\widetilde{H}$ is

$$PM(H, \widetilde{H}) = \sum_{\{\Gamma \mid \exists J\in[H]_\simeq, \widetilde{J}\in[\widetilde{H}]_\simeq\ J \xrightarrow{\Gamma} \widetilde{J}\}} PT(\Gamma, J).$$

Since $PM(H, \widetilde{H})$ is a probability for *any* multiset of activities to change H by $\widetilde{H}$, we use summation in the definition. Note that for every H holds $\sum_{\{\widetilde{H}|H\rightarrow\widetilde{H}\}} PM(H, \widetilde{H}) = 1$. This follows from the fact that $\sum_{\{\widetilde{H}|H\rightarrow\widetilde{H}\}} PM(H, \widetilde{H}) = \sum_{\{\widetilde{H}|H\rightarrow\widetilde{H}\}} \sum_{\{\Gamma\in Exec(H)|H\xrightarrow{\Gamma}\widetilde{H}\}} PT(\Gamma, H) = \sum_{\Gamma\in Exec(H)} PT(\Gamma, H) = 1$.

Definition 3.2 *Let G be a dynamic expression. The* (labeled probabilistic) transition system *of G is a quadruple*
$TS(G) = (S_G, L_G, \Omega_G, s_G)$, *where*

- *the set of* states *is $S_G = DR(G)$;*
- *the set of* labels *is $L_G = I\!N_f^{\mathcal{SL}}$;*
- *the* transition probability function *is $\Omega_G : S_G \times L_G \times S_G \rightarrow [0; 1]$ defined as follows: $\Omega_G([H]_\simeq, \Gamma, [\widetilde{H}]_\simeq) = PT(\Gamma, H)$, if $[H]_\simeq \in DR(G)$, $H \xrightarrow{\Gamma} \widetilde{H}\}$, otherwise, it is equal to zero;*
- *the* initial state *is $s_G = [G]_\simeq$.*

Thus, the transition system $TS(G)$ associated with a dynamic expression G describes all steps that happen at discrete moments of time with some probability and consist of multisets of activities. These steps change states, and the states are the isomorphism classes of dynamic expressions obtained by application of action rules starting from the expressions belonging to $[G]_\simeq$. If $\Omega_G(s, \Gamma, \tilde{s}) = \mathcal{P}$, the corresponding transition is written as $s \xrightarrow{\Gamma}_{\mathcal{P}} \tilde{s}$. This notation indicates that the probability to change the state s by $\tilde{s}$ as a result of executing Γ is $\mathcal{P}$.

We write $s \xrightarrow{\Gamma} \tilde{s}$ if $\exists\mathcal{P} > 0$ $s \xrightarrow{\Gamma}_{\mathcal{P}} \tilde{s}$. For one-element multiset $\Gamma = \{(\alpha, \rho)\}$ we write $s \xrightarrow{(\alpha,\rho)}_{\mathcal{P}} \tilde{s}$ and $s \xrightarrow{(\alpha,\rho)} \tilde{s}$.

Note that Γ could be the emptyset, and its execution does not change equivalence classes. This corresponds to the application of inaction rules to the expressions from the equivalence classes. We have to keep track of such executions called *empty loops*, because they have nonzero probabilities.

Transition systems of static expressions are defined in the following way. For $E \in StatExpr$ let $TS(E) = TS(\overline{E})$.

Definition 3.3 *Two dynamic expressions G and G' are* isomorphic w.r.t. transition systems, *denoted by $G =_{ts} G'$, if $TS(G) \simeq TS(G')$.*

Definition 3.4 *Let G be a dynamic expression. The* underlying discrete time Markov chain (DTMC) *of G, denoted by $DTMC(G)$, has the state space $DR(G)$ and transitions $[H]_\simeq \rightarrow_{PM(H,\widetilde{H})} [\widetilde{H}]_\simeq$, if $\exists\Gamma$ $[H]_\simeq \xrightarrow{\Gamma} [\widetilde{H}]_\simeq$.*

Note that for a dynamic expression G and $[H]_\simeq \in DR(G)$ we have $PM(H, \widetilde{H}) = \sum_{\{\Gamma|[H]_\simeq\xrightarrow{\Gamma}_{\mathcal{P}}[\widetilde{H}]_\simeq\}} \mathcal{P}$, i.e., the probability of each $DTMC(G)$ transition from a state s to $\tilde{s}$ is a sum of probabilities of $TS(G)$ transitions from s to $\tilde{s}$.

Example 3.1 *Let $E_1 = (\{a\}, \rho)[](\{a\}, \rho)$, $E_2 = (\{b\}, \chi)$ and $E = E_1; E_2$. The identical activities of the composite static expression are enumerated as follows: $E = ((\{a\}, \rho)_1[](\{a\}, \rho)_2); (\{b\}, \chi)$. In Figure 1 the transition system $TS(\overline{E})$ and the underlying DTMC $DTMC(\overline{E})$ are presented. Note that for the reason of simplicity in the graphical representation states are depicted by expressions belonging to the corresponding equivalence classes, and singleton multisets of activities are written without braces.*

4 Denotational semantics

In this section we construct denotational semantics in terms of a subclass of labeled DTSPNs called discrete time stochastic Petri boxes (dts-boxes). Since we propose stochastic extension of finite part of PBC, the dts-boxes will have finite observable behaviour.

4.1 Labeled DTSPNs

Now we introduce a class of labeled discrete time stochastic Petri nets.

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

- *P_N and T_N are finite sets of* places *and* transitions, *respectively, such that $P_N \cup T_N \neq \emptyset$ and $P_N \cap T_N = \emptyset$;*

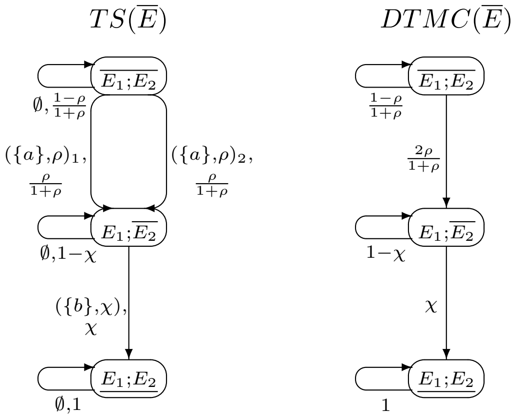

Figure 1: The transition system and the underlying DTMC of $\overline{E} = \overline{((\{a\},\rho)_1[](\{a\},\rho)_2);(\{b\},\chi)}$

- $W_N : (P_N \times T_N) \cup (T_N \times P_N) \to I\!N$ *is a function describing the* weights of arcs *between places and transitions and vice versa;*
- $\Omega_N : T_N \to (0;1)$ *is the* transition probability *function;*
- $L_N : T_N \to Act_\tau$ *is the* transition labeling *function assigning labels from a finite set of visible actions Act or an invisible action τ to transitions (i.e., $Act_\tau = Act \cup \{\tau\}$);*
- $M_N \in I\!N_f^{P_N}$ *is the* initial marking.

A graphical representation of LDTSPNs is as that for standard labeled Petri nets but with conditional probabilities written near the corresponding transitions. In the case the probabilities are not specified in the picture, they are considered to be of no importance in the corresponding examples, such as those used to describe stationary behaviour. The names of places and transitions are depicted near them when needed. If the names are omitted but used, it is supposed that the places and transitions are numbered from left to right and from top to down.

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

A transition $t \in T_N$ is enabled in a marking $M \in I\!N_f^{P_N}$ of LDTSPN N if ${}^\bullet t \subseteq M$. Let $Ena(M)$ be the set of *all transitions that are 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 sets (not multisets) of transitions may fire. Thus, we do not allow self-concurrency, i.e., firing of transitions concurrently to themselves. This restriction is because we would like to avoid technical difficulties while calculating probabilities for multisets of transitions as we shall see after the following formal definitions.

Let M be a marking of an LDTSPN N. A transition $t \in Ena(M)$ fires with *conditional* probability $\Omega_N(t)$ when no other transitions conflicting with it are enabled. Let $U \subseteq Ena(M)$. The *conditional* probability that the transitions from U fire (the case when no transitions conflicting with those from U are enabled) is

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

Thus, $PF(U,M)$ could be interpreted as a *joint* probability of independent events. Each such an event is interpreted as firing or not firing of a particular transition from U. The multiplication in the definition is because it reflects the probability of event intersection.

Let U be a transition set that is enabled in M. Concurrent firing of the transitions from U changes marking M by $\widetilde{M} = M - {}^\bullet U + U^\bullet$, denoted by $M \xrightarrow{U}_{\mathcal{P}} \widetilde{M}$. The probability of this step $\mathcal{P} = PT(U,M)$ is

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

Thus, $PT(U,M)$ is the probability that the set U fires *normalized* by the probability that *any* enabled in M set fires. The denominator of the fraction above is a summation since it reflects the probability of the event

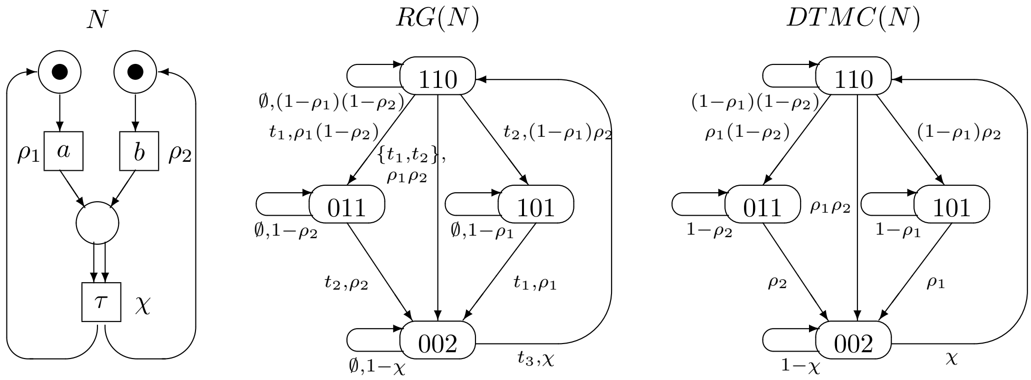

Figure 2: LDTSPN, its reachability graph and the underlying DTMC

union. The definition of $PT(U,M)$, unlike that of $PF(U,M)$, respects the fact that some transitions from $Ena(M)$ could be in conflict and hence cannot be fired together in some set.

We write $M \xrightarrow{U} \widetilde{M}$ if $\exists \mathcal{P} > 0\ M \xrightarrow{U}_{\mathcal{P}} \widetilde{M}$. For one-element transition set $U = \{t\}$ we write $M \xrightarrow{t}_{\mathcal{P}} \widetilde{M}$ and $M \xrightarrow{t} \widetilde{M}$.

Let us note that for all markings of an LDTSPN N the sum of outgoing probabilities is equal to one. More formally, $\forall M \in I\!N_f^{P_N}$ such that $Ena(M) \neq \emptyset$ we have $\sum_{\{U \subseteq Ena(M) \mid {}^\bullet U \subseteq M\}} PT(U,M) = 1$. This obviously follows from the definition of $PT(U,M)$ and guarantees that it defines a probability distribution.

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 $M \xrightarrow{U} \widetilde{M}$ then $\widetilde{M} \in RS(N)$.*
- *The* reachability graph *of N, denoted by $RG(N)$, is a directed labeled graph with the set of nodes $RS(N)$ and an arc labeled with $(U,\mathcal{P})$ between nodes M and $\widetilde{M}$ if $\exists \mathcal{P} > 0\ M \xrightarrow{U}_{\mathcal{P}} \widetilde{M}$.*
- *The* underlying discrete time Markov chain (DTMC) *of N, denoted by $DTMC(N)$, has the state space $RS(N)$ and transitions $M \rightarrow_{PM(M,\widetilde{M})} \widetilde{M}$, if $\exists U\ M \xrightarrow{U} \widetilde{M}$, where the transition probability is*

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

Thus, $PM(M,\widetilde{M})$ is a probability for *any* transition set to change marking M by $\widetilde{M}$, hence we use summation in the definition. Note that for every marking M holds $\sum_{\{\widetilde{M} \mid M \rightarrow \widetilde{M}\}} PM(M,\widetilde{M}) = 1$. This follows from the fact that $\sum_{\{\widetilde{M} \mid M \rightarrow \widetilde{M}\}} PM(M,\widetilde{M}) = \sum_{\{\widetilde{M} \mid M \rightarrow \widetilde{M}\}} \sum_{\{U \subseteq Ena(M) \mid M \xrightarrow{U} \widetilde{M}\}} PT(U,M) =$
$\sum_{\{U \subseteq Ena(M) \mid {}^\bullet U \subseteq M\}} PT(U,M) = 1$.

Example 4.1 *In Figure 2 an LDTSPN with two visible transitions t_1 (labeled by a), t_2 (labeled by b) and and one invisible transition t_3 (labeled by τ) is depicted. Transition probabilities of N are denoted by $\rho_i = \Omega_N(t_i)$ $(1 \leq i \leq 3)$. In the figure one can see the reachability graph $RG(N)$ and the underlying DTMC $DTMC(N)$ as well. The reachability set consists of markings $M_1 = (1,1,0)$, $M_2 = (0,1,1)$, $M_3 = (1,0,1)$, $M_4 = (0,0,2)$.*

4.2 Algebra of dts-boxes

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

Definition 4.3 *A* plain discrete time stochastic Petri box (plain dts-box) *is a tuple $N = (P_N, T_N, W_N, \Lambda_N)$, where*

- P_N *and* T_N *are finite sets of* places *and* transitions*, respectively, 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 describing the* weights of arcs *between places and transitions and vice versa;*

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

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

A *marked plain dts-box* is a pair (N, M_N), where N is a plain dts-box and $M_N \in I\!N_f^{P_N}$ is the initial marking. Note that a marked plain dts-box $(P_N, T_N, W_N, \Lambda_N, M_N)$ could be interpreted as the LDTSPN $(P_N, T_N, W_N, \Omega_N, L_N, M_N)$, where functions Ω_N and L_N are defined as follows: $\forall t \in T_N \ \Omega_N(t) = \Omega(\Lambda_N(t))$, $L_N(t) = \mathcal{L}(\Lambda_N(t))$. In this case, the label τ of silent transitions from the LDTSPN corresponds to the multiaction part $\emptyset$ of activities that label unobservable transitions of the the corresponding dts-box. The behaviour of marked dts-boxes follows to the firing rule of LDTSPNs. A plain dts-box N is *safe*, if $(N, {}^\bullet N)$ is, i.e., $\forall M \in RS(N, {}^\circ N) \ M \subseteq P_N$. A safe plain dts-box N is *clean* if $N^\circ \subseteq M \ \Rightarrow \ M = N^\circ$, i.e., if there are tokens in exit places then all and only exit places have tokens.

To define semantic function that associates a plain dts-box with every static expression of *dtsPBC*, we need to propose the *enumeration* function $Enu : T_N \to I\!N^*$. It imposes the numbers with transitions of plain dts-box N in accordance to the enumeration of activities from left to right in the syntax of the underlying static expression. In the case of synchronization, the function associates the concatenation of the numbers of the transitions it comes from with the resulting new transition.

The structure of the plain dts-box corresponding to a static expression is constructed as in *PBC*, i.e., via refinement and labeling. Thus, the resulting dts-boxes are safe and clean. In the definition of denotational semantics we shall use standard constructions used for *PBC* in [12, 13, 6]. For convenience, we only use slightly different notation: ϱ, Θ and u stand for ρ, Ω and v from *PBC* setting, respectively.

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

- $\varrho_{id} = \{(\{(\alpha, \rho)\}, (\alpha, \rho) \mid (\alpha, \rho) \in \mathcal{SL}\}$;

- $\varrho_{[f]} = \{(\{(\alpha, \rho)\}, (f(\alpha), \rho) \mid (\alpha, \rho) \in \mathcal{SL}\}$;

- $\varrho_{\mathsf{rs}\ a} = \{(\{(\alpha, \rho)\}, (\alpha, \rho) \mid (\alpha, \rho) \in \mathcal{SL},\ a, \hat{a} \not\in \mathcal{A}(\alpha)\}$;

- $\varrho_{\mathsf{sy}\ a}$ is the least relabeling relation contained in ϱ_{id} such that if $(\Gamma, \{(\alpha + \{a\}, \rho)\} \in \varrho_{\mathsf{sy}\ a}$ and $(\Delta, \{(\beta + \{\hat{a}\}, \chi)\} \in \varrho_{\mathsf{sy}\ a}$ then $(\Gamma + \Delta, \{(\alpha + \beta, \rho \cdot \chi)\} \in \varrho_{\mathsf{sy}\ a}$.

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

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

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

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

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

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

$$Enu(t) = Enu_E(t),\ t \in T_E,\ a, \hat{a} \not\in L_E(t).$$

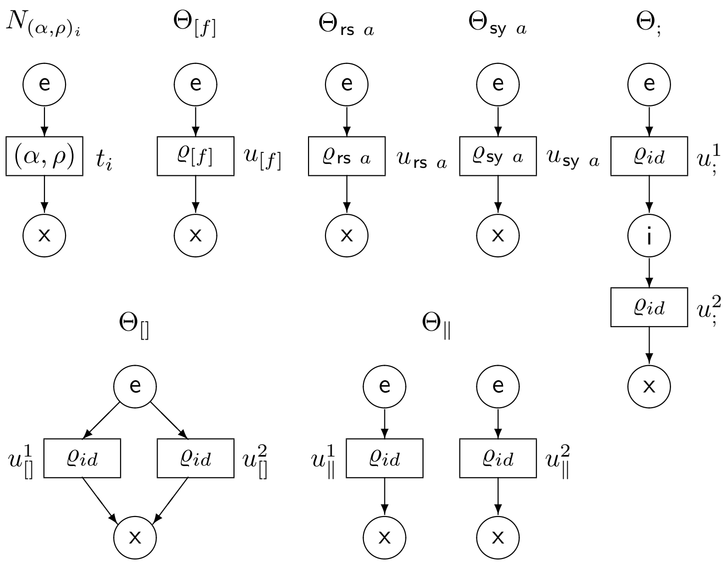

Figure 3: The plain and operator dts-boxes

- $Box_{dts}(E\ \mathsf{sy}\ a) = \Theta_{\mathsf{sy}\ a}(Box_{dts}(E))$. Note that $\forall v, w \in T_E$ such that $L_E(v) = \alpha + \{a\},\ L_E(w) = \beta + \{\hat{a}\}$, the new transition t resulting from synchronization of v and w has label $L(t) = \alpha + \beta$, conditional probability $\Omega(t) = \Omega_E(v) \cdot \Omega_E(w)$ and enumeration $Enu(t) = Enu_E(v) \cdot Enu_E(w)$. Thus, the enumeration is defined as

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

  To avoid introducing redundant transitions generated by synchronizing in different order the same transition set, we only consider a single one of them in the plain dts-box.

The plain and operator dts-boxes are presented in Figure 3.
Now we can formally define denotational semantics as a homomorphism.

Definition 4.4 *Let $(\alpha, \rho) \in \mathcal{SL}$ and $E, F, \in StatExpr$. The* denotational semantics *$dtsPBC$ is a mapping Box_{dts} from $StatExpr$ into the area of plain dts-boxes defined as follows:*

1. $Box_{dts}((\alpha, \rho)_i) = N_{(\alpha,\rho)_i}$;
2. $Box_{dts}(E \circ F) = \Theta_\circ(Box_{dts}(E), Box_{dts}(F)),\ \circ \in \{;, [], \|\}$;
3. $Box_{dts}(E[f]) = \Theta_{[f]}(Box_{dts}(E))$;
4. $Box_{dts}(E \circ a) = \Theta_{\circ a}(Box_{dts}(E)),\ \circ \in \{\mathsf{rs}, \mathsf{sy}\}$.

Isomorphism is a coincidence of systems up to renaming of their components or states. We denote isomorphism of transition systems by $\simeq$, and the same symbol denotes isomorphism of reachability graphs and DTMCs. Moreover, $\simeq$ will denote an isomorphism between transition systems and reachability graphs. Note that in this case, the names of transitions of the dts-box corresponding to a static expression could be identified with the enumerated activities of the latter.

Theorem 4.1 *For any static expression E*

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

Proof. What concerns qualitative (functional) behaviour, we have the same isomorphism as in PBC.
The quantitative behaviour is equal by the following reasons. First, the activities of a static expression have probability parts coinciding with conditional probabilities of the transitions belonging to the corresponding plain dts-box. Second, in both semantics conflicts are resolved via the same probability functions. $\square$

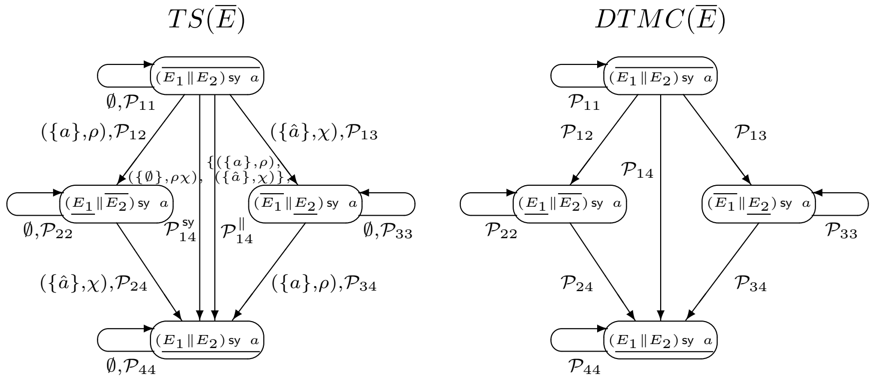

Figure 4: The transition system and the underlying DTMC of $\overline{E} = \overline{((\{a\},\rho)\|(\{\hat{a}\},\chi))\ \mathsf{sy}\ a}$

Proposition 4.1 *For any static expression E*

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

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

The dts-boxes of dynamic expressions can be defined as well. For $E \in StatExpr$ let $Box_{dts}(\overline{E}) = \overline{Box_{dts}(E)}$ and $Box_{dts}(\underline{E}) = \underline{Box_{dts}(E)}$. Note that any dynamic expression can be decomposed into pure, overlined or underlined static expressions, and the definition of dts-boxes is compositional.

Example 4.2 *Let $E_1 = (\{a\},\rho)$, $E_2 = (\{\hat{a}\},\chi)$ and $E = (E_1\|E_2)\ \mathsf{sy}\ a = ((\{a\},\rho)\|(\{\hat{a}\},\chi))\ \mathsf{sy}\ a$. In Figure 4 the transition system $TS(\overline{E})$ and the underlying DTMC $DTMC(\overline{E})$ are presented. In Figure 5 the marked dts-box $N = (Box_{dts}(E), {}^{\circ}Box_{dts}(E))$, its reachability graph $RG(N)$ and the underlying DTMC $DTMC(N)$ are presented. It is easy to see that $TS(\overline{E})$ and $RG(N)$ are isomorphic as well as $DTMC(\overline{E})$ and $DTMC(N)$.*

The 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 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:

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

5 Probabilistic equivalences

In this section we propose a number of probabilistic equivalences of expressions. Semantic equivalence $=_{ts}$ is too strict in many cases, hence, we need weaker equivalence notions to compare behaviour of processes specified by algebraic formulas.

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

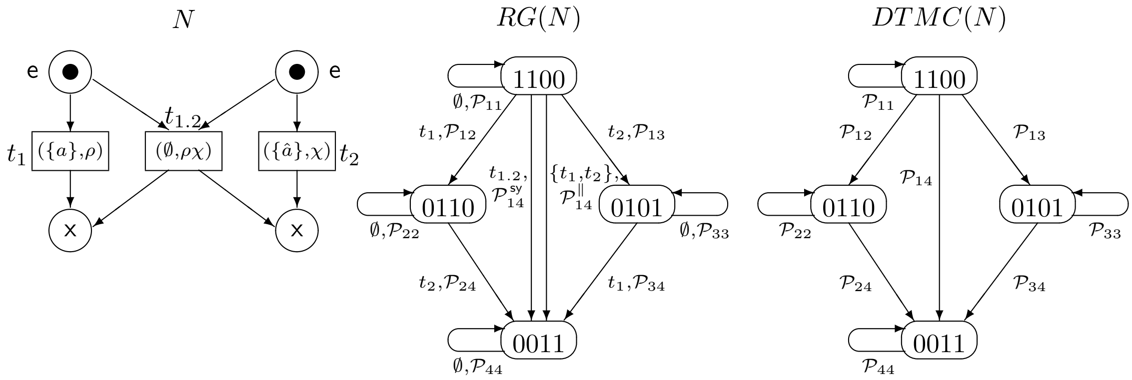

Figure 5: The marked dts-box $N = (Box_{dts}(E), {}^{\circ}Box_{dts}(E))$, its reachability graph and the underlying DTMC

5.1 Empty loops in transition systems

Let G be a dynamic expression. Transition system $TS(G)$ can have loops going from a state to itself which are labeled by an emptyset and have non-zero probability. Such the *empty loop* $s \xrightarrow{\emptyset}_{\mathcal{P}} s,\ \mathcal{P} > 0$ appears when no activities occur at a time step, and this happens with some positive probability. Obviously, in this case the current state remains unchanged.

Let G be a dynamic expression and $[H]_{\simeq} \in DR(G)$.

The *probability to stay in* $[H]_{\simeq}$ *due to* k $(k \geq 0)$ *empty loops* is

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

The *probability to execute in* $[H]_{\simeq}$ *a non-empty multiset of activities* Γ *after possible empty loops* is

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

Definition 5.1 *The* (labeled probabilistic) transition system without empty loops $TS^*(G)$ *has the state space* $DR(G)$ *and the transitions* $[H]_{\simeq} \overset{\Gamma}{\twoheadrightarrow}_{PT^*(\Gamma,H)} [\widetilde{H}]_{\simeq}$*, if* $[H]_{\simeq} \xrightarrow{\Gamma} [\widetilde{H}]_{\simeq},\ \Gamma \neq \emptyset$.

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 $\exists \mathcal{P} > 0\ s \overset{\Gamma}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s}$. For one-element transition set $\Gamma = \{(\alpha, \rho)\}$ we write $s \overset{(\alpha,\rho)}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s}$ and $s \overset{(\alpha,\rho)}{\twoheadrightarrow} \tilde{s}$.

We decided to consider only an empty loop followed by a non-empty step just for convenience. Alternatively, we could consider a non-empty step succeeded by an empty loop or a non-empty step preceded and succeeded by empty loops. In both cases our sequence begins or/and ends with loops that do not change states. Only overall probabilities of these three 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 that resembles in some sense a construction of branching bisimulation [27].

Transition systems without empty loops of static expressions are defined in the following way. For $E \in StatExpr$ let $TS^*(E) = TS^*(\overline{E})$.

Definition 5.2 *Two dynamic expressions* G *and* G' *are* isomorphic w.r.t. transition systems without empty loops*, denoted by* $G =_{ts*} G'$*, if* $TS^*(G) \simeq TS^*(G')$.

Definition 5.3 *The* underlying DTMC without empty loops $DTMC^*(G)$ *has the state space* $DR(G)$ *and transitions* $[H]_{\simeq} \twoheadrightarrow_{PM^*(H,\widetilde{H})} [\widetilde{H}]_{\simeq}$*, if* $\exists \Gamma\ [H]_{\simeq} \xrightarrow{\Gamma} [\widetilde{H}]_{\simeq}$*, where the transition probability is*

$$PM^*(H, \widetilde{H}) = \sum_{\{\Gamma \mid [H]_{\simeq} \overset{\Gamma}{\twoheadrightarrow} [\widetilde{H}]_{\simeq}\}} PT^*(\Gamma, H).$$

Let us note that $PT^*(\Gamma, H)$ defines a probability distribution, i.e., $\forall H \in DR(G)\ \sum_{\Gamma \in Exec(H)\setminus\emptyset} PT^*(\Gamma, H) = 1$.

In some cases, interleaving behaviour is to be considered. Interleaving semantics abstracts from steps with more than one element. After such an abstracting, one has to normalize probabilities of the remaining one-element steps. We need to do it since the sum of outgoing probabilities should always be equal to one for each marking to form a probability distribution. For this, a special *interleaving transition relation* is proposed. Let G be a dynamic expression and $s, \tilde{s} \in DR(G),\ (\alpha, \rho) \in Exec(H)$. We write $s \xrightarrow{(\alpha,\rho)}_{\mathcal{Q}} \tilde{s}$ if $s \xrightarrow{(\alpha,\rho)}_{\mathcal{P}} \tilde{s}$ and

$$\mathcal{Q} = \frac{\mathcal{P}}{\sum_{\{(\beta,\chi)\in Exec(H)),\ \tilde{s}\in DR(G) \mid s \xrightarrow{(\beta,\chi)}_{\mathcal{P}'} \tilde{s}\}} \mathcal{P}'}.$$

5.2 Empty loops in reachability graphs

Let N be an LDTSPN and M be its marking. Reachability graph $RG(N)$ can have loops going from a state to itself which are labeled by an emptyset and have non-zero probability. Such the *empty loop* $M \xrightarrow{\emptyset}_{\mathcal{P}} M,\ \mathcal{P} > 0$ appears when no transitions fire at a time step, and this happens with some positive probability. Obviously, in this case the current marking remains unchanged.

The *probability to stay in* M *due to* k $(k \geq 0)$ *empty loops* is

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

The *probability to execute in* M *a non-empty transition set* U *after possible empty loops* is

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

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

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

We write $M \overset{U}{\twoheadrightarrow} \widetilde{M}$ if $\exists \mathcal{P} > 0\ M \overset{U}{\twoheadrightarrow}_{\mathcal{P}} \widetilde{M}$. For one-element transition set $U = \{t\}$ we write $M \overset{t}{\twoheadrightarrow}_{\mathcal{P}} \widetilde{M}$ and $M \overset{t}{\twoheadrightarrow} \widetilde{M}$.

We decided to consider only an empty loop followed by a non-empty step just for convenience. Alternatively, we could consider a non-empty step succeeded by an empty loop or a non-empty step preceded and succeeded by empty loops. In both cases our sequence begins or/and ends with loops that do not change markings. Only overall probabilities of these three 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 that resembles in some sense a construction of branching bisimulation [27].

Definition 5.5 *The* underlying DTMC without empty loops $DTMC^*(N)$ *has the state space* $RS(N)$ *and transitions* $M \twoheadrightarrow_{PM^*(M,\widetilde{M})} \widetilde{M}$*, if* $\exists U\ M \xrightarrow{U} \widetilde{M}$*, where the transition probability is*

$$PM^*(M, \widetilde{M}) = \sum_{\{U \in Ena(M) \mid M \overset{U}{\twoheadrightarrow} \widetilde{M}\}} PS^*(U, M).$$

Note that $PT^*(U, M)$ defines a probability distribution, i.e., $\forall M \in RS(N)\ \sum_{U \in Ena(M)\setminus\emptyset} PT^*(U, M) = 1$.

In some cases, interleaving behaviour is to be considered. Interleaving semantics abstracts from steps with more than one element. After such an abstracting, one has to normalize probabilities of the remaining one-element steps. For this, a special *interleaving transition relation* is proposed. Let N be an LDTSPN and $M, \widetilde{M} \in RS(N),\ t \in Ena(M)$. We write $M \overset{t}{\twoheadrightarrow}_{\mathcal{Q}} \widetilde{M}$ if $M \overset{t}{\twoheadrightarrow}_{\mathcal{P}} \widetilde{M}$ and

$$\mathcal{Q} = \frac{\mathcal{P}}{\sum_{\{u \in Ena(M),\ \widetilde{M} \in RS(N) \mid M \overset{u}{\twoheadrightarrow}_{\mathcal{P}'} \widetilde{M}\}} \mathcal{P}'}.$$

Theorem 5.1 *For any static expression* E

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

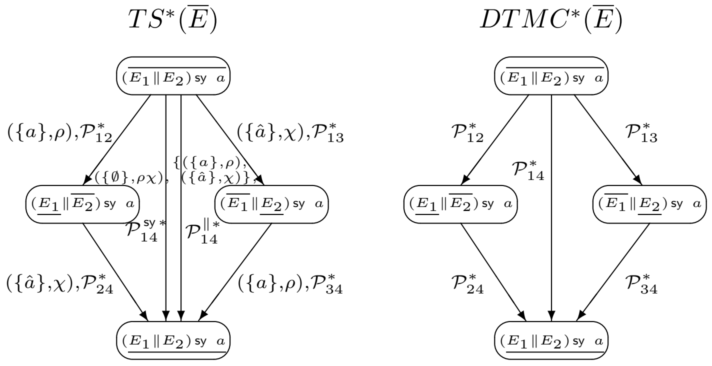

Figure 6: The transition system and the underlying DTMC without empty loops of $\overline{E}$ from Example 4.2

Proof. As Theorem 4.1. □

Proposition 5.1 *For any static expression E*

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

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 empty loops free transition system of a dynamic expression, a similarly defined analogue exists on the empty loops free reachability graph of the corresponding dts-box.

Example 5.1 *Let E and N be those from Example 4.2. In Figure 6 the transition system $TS^*(\overline{E})$ and the underlying DTMC $DTMC^*(\overline{E})$ without empty loops are presented. In Figure 7 the reachability graph $RG^*(N)$ and the underlying DTMC $DTMC^*(N)$ without from empty loops are presented. It is easy to see that $TS^*(\overline{E})$ and $RG^*(N)$ are isomorphic as well as $DTMC^*(\overline{E})$ and $DTMC^*(N)$.*

The 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 to the concurrent execution of two activities. To avoid complex notation, we use the normalization factor $\mathcal{N}^* = \frac{1}{\rho+\chi-2\rho^2\chi-2\rho\chi^2+2\rho^2\chi^2}$. Note that the probabilities $\mathcal{P}_{ij}$ $(1 \le i, j \le 4)$ are taken from Example 4.2.*

$$\mathcal{P}^*_{12} = \frac{\mathcal{P}_{12}}{1-\mathcal{P}_{11}} = \mathcal{N}^*\rho(1-\chi)(1-\rho\chi) \qquad \mathcal{P}^*_{13} = \frac{\mathcal{P}_{13}}{1-\mathcal{P}_{11}} = \mathcal{N}^*\chi(1-\rho)(1-\rho\chi)$$

$$\mathcal{P}^{\mathsf{sy}\,*}_{14} = \frac{\mathcal{P}^{\mathsf{sy}}_{14}}{1-\mathcal{P}_{11}} = \mathcal{N}^*\rho\chi(1-\rho)(1-\chi) \qquad \mathcal{P}^{\|*}_{14} = \frac{\mathcal{P}^{\|}_{14}}{1-\mathcal{P}_{11}} = \mathcal{N}^*\rho\chi(1-\rho\chi)$$

$$\mathcal{P}^*_{24} = \frac{\mathcal{P}_{24}}{1-\mathcal{P}_{22}} = 1 \qquad \mathcal{P}^*_{34} = \frac{\mathcal{P}_{34}}{1-\mathcal{P}_{33}} = 1$$

$$\mathcal{P}^*_{14} = \mathcal{P}^{\mathsf{sy}\,*}_{14} + \mathcal{P}^{\|*}_{14} = \frac{\mathcal{P}^{\mathsf{sy}}_{14}+\mathcal{P}^{\|}_{14}}{1-\mathcal{P}_{11}} = \mathcal{N}^*\rho\chi(2-\rho-\chi)$$

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

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

5.3 Probabilistic trace equivalences

Trace equivalences are the least distinctive ones. In the trace semantics, 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 probabilistic trace relations resemble those of trace equivalences for standard Petri nets [63] or process algebras, but additionally we have to take into account the probabilities of sequences of (multisets of) multiactions. 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.

For $\Gamma \in I\!N_f^{\mathcal{SL}}$, we define its *multiaction part* by $\mathcal{L}(\Gamma) = \sum_{(\alpha,\rho)\in\Gamma} \alpha$. Note that $\mathcal{L}(\Gamma) \in I\!N_f^{\mathcal{L}}$, i.e, $\mathcal{L}(\Gamma)$ is a multiset of multiactions.

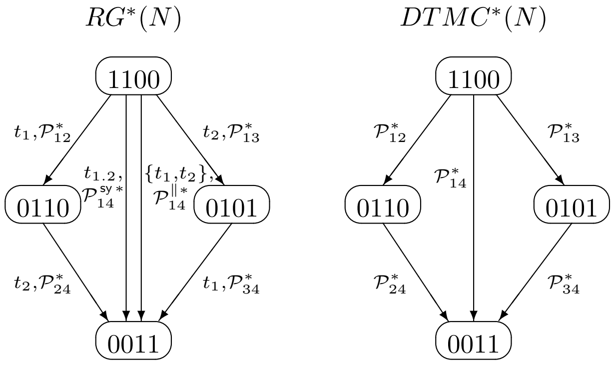

Figure 7: The reachability graph and the underlying DTMC without empty loops of N from Example 4.2

Definition 5.6 *An* interleaving probabilistic trace *of a dynamic expression G with $TS(G) = (S_G, L_G, \Omega_G, s_G)$ is a pair $(\sigma, \mathcal{P})$, where $\sigma = \alpha_1 \cdots \alpha_n \in \mathcal{L}^*$ and*

$$\mathcal{P} = \sum_{\{(\alpha_1,\rho_1),\ldots,(\alpha_n,\rho_n) \mid s_G \overset{(\alpha_1,\rho_1)}{\twoheadrightarrow}_{\mathcal{P}_1} s_1 \overset{(\alpha_2,\rho_2)}{\twoheadrightarrow}_{\mathcal{P}_2} \cdots \overset{(\alpha_n,\rho_n)}{\twoheadrightarrow}_{\mathcal{P}_n} s_n\}} \prod_{i=1}^{n} \mathcal{P}_i.$$

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

$$IntProbTraces(G) = IntProbTraces(G').$$

Definition 5.7 *A* step probabilistic trace *of a dynamic expression G with $TS(G) = (S_G, L_G, \Omega_G, s_G)$ is a pair $(\Sigma, \mathcal{P})$, where $\Sigma = A_1 \cdots A_n \in (I\!N_f^{\mathcal{L}})^*$ and*

$$\mathcal{P} = \sum_{\{\Gamma_1,\ldots,\Gamma_n \mid s_G \overset{\Gamma_1}{\twoheadrightarrow}_{\mathcal{P}_1} s_1 \overset{\Gamma_2}{\twoheadrightarrow}_{\mathcal{P}_2} \cdots \overset{\Gamma_n}{\twoheadrightarrow}_{\mathcal{P}_n} s_n,\ \mathcal{L}(\Gamma_i)=A_i\ (1 \le i \le n)\}} \prod_{i=1}^{n} \mathcal{P}_i.$$

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

$$StepProbTraces(G) = StepProbTraces(G').$$

5.4 Probabilistic bisimulation equivalences

Bisimulation equivalences respect completely the particular points of choice in the behavior of a modeled system. We intend to present a parameterized definition of probabilistic bisimulation equivalences.

To define probabilistic bisimulation equivalences, we have to consider a bisimulation as an *equivalence* relation which 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, s_G and $s_{G'}$, 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, sums of probabilities for all such occurrences should be the same for both states. Thus, in our definitions, we follow the approach of [39, 40]. 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*.

First, we introduce several helpful notations. Let for a dynamic expression G we have $\mathcal{H} \subseteq DR(G)$. Then for some $s \in DR(G)$ and $A \in I\!N_f^{\mathcal{L}}$ we write $s \overset{A}{\twoheadrightarrow}_{\mathcal{Q}} \mathcal{H}$ if

$$\mathcal{Q} = \sum_{\{\Gamma \mid s \overset{\Gamma}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s},\ \mathcal{L}(\Gamma)=A,\ \tilde{s} \in \mathcal{H}\}} \mathcal{P}.$$

We write $s \overset{A}{\twoheadrightarrow} \mathcal{H}$ if $\exists \mathcal{Q} > 0\ s \overset{A}{\twoheadrightarrow}_{\mathcal{Q}} \mathcal{H}$.

In similar way, we define the notion $s \xrightharpoonup{\alpha}_{\mathcal{Q}} \mathcal{H}$ based on the interleaving transition relation.

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 an *equivalence class* (w.r.t. $\mathcal{E}$) 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 in the *set of equivalence classes* $X/_{\mathcal{E}} = \{[x]_{\mathcal{E}} \mid x \in X\}$.

Definition 5.8 *Let G be a dynamic expression and $TS(G) = (S_G, L_G, \Omega_G, s_G)$ be its transition system. An* equivalence *relation $\mathcal{R} \subseteq DR(G)^2$ is a* $\star$-probabilistic bisimulation *between states s_1 and s_2 of $TS(G)$,* $\star \in$ {interleaving, step}*, denoted by $\mathcal{R} : s_1 \underline{\leftrightarrow}_{\star p} s_2,\ \star \in \{i, s\}$, if $\forall \mathcal{H} \in DR(G)/_{\mathcal{R}}$*

- *$\forall x \in \mathcal{L}$ and $\hookrightarrow = \rightharpoonup$, if $\star = i$;*
- *$\forall x \in I\!N_f^{\mathcal{L}}$ and $\hookrightarrow = \twoheadrightarrow$, if $\star = s$;*

$$s_1 \overset{x}{\hookrightarrow}_{\mathcal{Q}} \mathcal{H} \ \Leftrightarrow\ s_2 \overset{x}{\hookrightarrow}_{\mathcal{Q}} \mathcal{H}.$$

Two states s_1 and s_2 are $\star$-probabilistic bisimulation equivalent*,* $\star \in$ {interleaving, step}*, denoted by $s_1 \underline{\leftrightarrow}_{\star p} s_2$, if $\exists \mathcal{R} : s_1 \underline{\leftrightarrow}_{\star p} s_2,\ \star \in \{i, s\}$.*

To introduce bisimulation between dynamic expressions G and G', we should consider a "composite" set of states $DR(G) \cup DR(G')$.

Definition 5.9 *Let G, G' be dynamic expressions and $TS(G) = (S_G, L_G, \Omega_G, s_G)$, $TS(G') = (S_{G'}, L_{G'}, \Omega_{G'}, s_{G'})$ be their transition systems. A relation $\mathcal{R} \subseteq (DR(G) \cup DR(G'))^2$ is a* $\star$-probabilistic bisimulation *between G and G',* $\star \in$ {interleaving, step}*, denoted by $\mathcal{R} : G \underline{\leftrightarrow}_{\star p} G'$, if $\mathcal{R} : s_G \underline{\leftrightarrow}_{\star p} s_{G'},\ \star \in \{i, s\}$.*

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

5.5 Stochastic isomorphism

Stochastic isomorphism is a relation that is weaker than the equivalence with respect to the isomorphism of the associated transition systems without empty loops. The main idea of the following definition is to summarize 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 event union.

Definition 5.10 *Let G, G' be dynamic expressions and $TS(G) = (S_G, L_G, \Omega_G, s_G)$, $TS(G') = (S_{G'}, L_{G'}, \Omega_{G'}, s_{G'})$ be their transition systems. A mapping $\beta : S_G \to S_{G'}$ is a* stochastic isomorphism *between G and G', denoted by $\beta : G =_{sto} G'$, if*

1. *β is a bijection such that $\beta(s_G) = \beta(s_{G'})$;*
2. *if $s \overset{\Gamma}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s}$ then there exists Γ' such that $\beta(s) \overset{\Gamma'}{\twoheadrightarrow}_{\mathcal{P}'} \beta(\tilde{s}),\ \mathcal{L}(\Gamma) = \mathcal{L}(\Gamma')$ and*
$$\sum_{\{\Delta \mid s \overset{\Delta}{\twoheadrightarrow}_{\mathcal{Q}} \tilde{s},\ \mathcal{L}(\Gamma) = \mathcal{L}(\Delta)\}} \mathcal{Q} = \sum_{\{\Delta' \mid \beta(s) \overset{\Delta'}{\twoheadrightarrow}_{\mathcal{Q}'} \beta(\tilde{s}),\ \mathcal{L}(\Gamma) = \mathcal{L}(\Delta')\}} \mathcal{Q}';$$
3. *if $s' \overset{\Gamma'}{\twoheadrightarrow}_{G'} \tilde{s}'$ then there exists Γ such that $\beta^{-1}(s') \overset{\Gamma}{\twoheadrightarrow}_{G} \beta^{-1}(\tilde{s}'),\ \mathcal{L}(\Gamma) = \mathcal{L}(\Gamma')$ and*
$$\sum_{\{\Delta' \mid s' \overset{\Delta'}{\twoheadrightarrow}_{\mathcal{Q}'} \tilde{s}',\ \mathcal{L}(\Gamma) = \mathcal{L}(\Delta')\}} \mathcal{Q}' = \sum_{\{\Delta \mid \beta^{-1}(s') \overset{\Delta}{\twoheadrightarrow}_{\mathcal{Q}} \beta^{-1}(\tilde{s}')),\ \mathcal{L}(\Gamma) = \mathcal{L}(\Delta)\}} \mathcal{Q}.$$

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

$$\begin{tikzcd}
 & \simeq \arrow[d] \\
 & =_{ts} \arrow[d] \\
 & =_{ts*} \arrow[d] \\
 & =_{sto} \arrow[d] \\
\underline{\leftrightarrow}_{ip} \arrow[d] & \underline{\leftrightarrow}_{sp} \arrow[l] \arrow[d] \\
\equiv_{ip} & \equiv_{sp} \arrow[l]
\end{tikzcd}$$

Figure 8: Interrelations of the probabilistic equivalences

5.6 Interrelations of the probabilistic equivalences

Now we intend to compare the introduced probabilistic equivalences and obtain the lattice of their interrelations.

Proposition 5.2 *Let $\star \in \{i, s\}$. For dynamic expressions G and G' the following holds:*

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

Proof. See Appendix A. $\square$

In the following, the symbol '_' will denote "nothing", and the equivalences subscribed by it are considered as those without any subscription.

Theorem 5.2 *Let $\leftrightarrow, \twoheadleftarrow\!\!\twoheadrightarrow \in \{\equiv, \underline{\leftrightarrow}, =, \simeq\}$ and $\star, \star\star \in \{\_, ip, sp, sto, ts*, ts\}$. For dynamic expressions G and G'*

$$G \leftrightarrow_\star G' \;\Rightarrow\; G \twoheadleftarrow\!\!\twoheadrightarrow_{\star\star} G'$$

iff in the graph in Figure 8 there exists a directed path from $\leftrightarrow_\star$ to $\twoheadleftarrow\!\!\twoheadrightarrow_{\star\star}$.

Proof. ($\Leftarrow$) Let us check the validity of implications in the graph in Figure 8.

- The implications $\leftrightarrow_{sp} \to \leftrightarrow_{ip}$, $\leftrightarrow \in \{\equiv, \underline{\leftrightarrow}\}$ are valid, since single activities are one-element multisets.
- The implications $\underline{\leftrightarrow}_\star \to \equiv_\star$, $\star \in \{ip, sp\}$, are valid by Proposition 5.2.
- The implication $=_{sto} \to \underline{\leftrightarrow}_{sp}$ is proved as follows. Let $\beta : G =_{sto} G'$. Then it is easy to see that $\mathcal{S} : G \underline{\leftrightarrow}_{sp} G'$, where $\mathcal{S} = \{(s, \beta(s)) \mid s \in DR(G)\}$.
- The implication $=_{ts*} \to =_{sto}$ is valid, since stochastic isomorphism is that of empty loops free transition systems up to merging of transitions with labels having identical multiaction parts.
- The implication $=_{ts} \to =_{ts*}$ is valid, since abstraction from empty loops is based on transition probabilities which are the same for isomorphic transition systems.

- The implication $\simeq \to =_{ts}$ is valid, since the transition system of a dynamic formula is defined based on its isomorphism class.

($\Rightarrow$) Absence of additional nontrivial arrows in the graph in Figure 8 is proved by the following examples. As in the previous examples, we assume that conflicting transitions have equal weights and probabilities.

- Let $G = \overline{(\{a\}, \frac{1}{2}) \| (\{b\}, \frac{1}{2})}$ and $G' = \overline{((\{a\}, \frac{1}{2}); (\{b\}, \frac{1}{2}))[]((\{b\}, \frac{1}{2}); (\{a\}, \frac{1}{2}))}$. Then $G \underline{\leftrightarrow}_{ip} G'$, but $G \not\equiv_{sp} G'$, since only in $TS(G')$ multiactions $\{a\}$ and $\{b\}$ cannot be executed concurrently.
- Let $G = \overline{(\{a\}, \frac{1}{2}); ((\{b\}, \frac{1}{2})[](\{c\}, \frac{1}{2}))}$ and $G' = \overline{((\{a\}, \frac{1}{2}); (\{b\}, \frac{1}{2}))[]((\{a\}, \frac{1}{2}); (\{c\}, \frac{1}{2}))}$. Then $G \equiv_{sp} G'$, but $G \underline{\not\leftrightarrow}_{ip} G'$, since only in $TS(G')$ a multiaction $\{a\}$ can be executed so that no multiaction $\{b\}$ can occur afterwards.
- Let $G = \overline{(\{a\}, \frac{1}{2}); (\{b\}, \frac{1}{2})}$ and $G' = \overline{(\{a\}, \frac{1}{2}); (\{b\}, \frac{1}{2})[](\{a\}, \frac{1}{2}); (\{b\}, \frac{1}{2})}$. Then $G \underline{\leftrightarrow}_{sp} G'$, but $G \neq_{sto} G'$, since only in $TS(G')$ there is a transition with multiaction part of label $\{a\}$ and probability 1 that is single one between its start and final states such that the transition has no corresponding transition set in $TS(G')$. Note that in $TS(G')$, the only transition with the same multiaction part of label has probability $\frac{1}{2}$.
- Let $G = \overline{(\{a\}, \frac{1}{2})}$ and $G' = \overline{(\{a\}, \frac{1}{2})[](\{a\}, \frac{1}{2})}$. Then $G =_{sto} G'$, but $G \neq_{dts*} G'$, since only $TS(G')$ has two transitions.
- Let $G = \overline{(\{a\}, \frac{1}{2})}$ and $G = \overline{(\{a\}, \frac{1}{3})}$. Then $G =_{ts*} G'$, but $G \neq_{dts} G'$, since only in $TS(G')$ there is a transition with multiaction part of label $\{a\}$ and probability $\frac{1}{3}$.
- Let $G = \overline{(\{a\}, \frac{1}{2})}$ and $G' = \overline{(\{a\}, \frac{1}{2}); (\{\hat{a}\}, \frac{1}{2}) \text{ sy } a}$. Then $G =_{ts} G'$, but $G \not\simeq G'$, since G and G' cannot be reached each from other by applying inaction rules. □

Example 5.2 *In Figure 9 the marked dts-boxes corresponding to the dynamic expressions from equivalence examples of Theorem 5.2 are presented, i.e.,* $\overline{E} = G$, $N = (Box_{dts}(E), {}^{\circ}Box_{dts}(E))$ *and* $\overline{E'} = G'$, $N' = (Box_{dts}(E'), {}^{\circ}Box_{dts}(E'))$ *for each picture (a)–(f). Since all the equivalences of dynamic expressions can be transferred to the corresponding marked dts-boxes, we depict also which the net analogues (denoted by the same symbols) of the algebraic equivalences relate the nets.*

6 Conclusion

In this paper, we have proposed a discrete time stochastic extension of PBC called $dtsPBC$ with concurrent step operational semantics based on transition systems and denotational semantics in terms of a subclass of LDTSPNs. An accordance of operational and denotational semantics was established. In addition, we have defined a number of probabilistic 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 was constructed.

Future work consists in the construction a congruence relation based on some probabilistic algebraic equivalence we defined. We can also abstract from the silent activities, i.e., those with empty multiaction part in the definitions of the equivalences. The abstraction from empty loops and that from silent activities could be done in one step as well. The main point is that we should collect probabilities during such the abstractions from the internal activity. As a result, we shall have the algebraic analogues of the net probabilistic equivalences from [15, 16]. Moreover, we plan to extend $dtsPBC$ with infiniteness constructs such as iteration and recursion.

References

[1] ANDOVA S. *Process algebra with probabilistic choice. Lecture Notes in Computer Science* **1601**, p. 111–129, 1999.

[2] 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.

[3] 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.

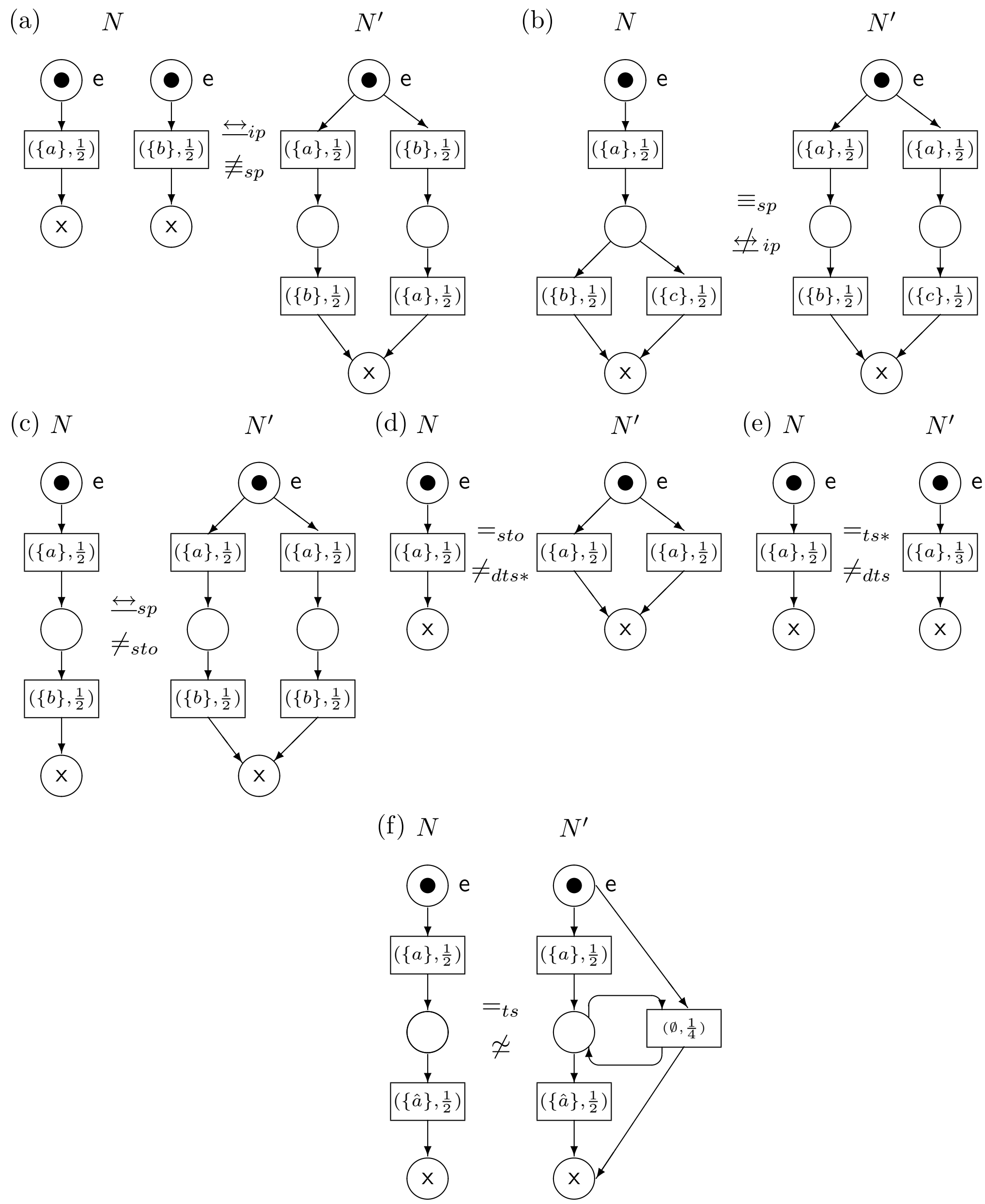

Figure 9: Dts-boxes of the dynamic expressions from equivalence examples of Theorem 5.2

[4] 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.

[5] 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.

[6] Best E., Devillers R., Koutny M. *Petri net algebra. EATCS Monographs on Theoretical Computer Science*, 378 p., Springer Verlag, 2001.

[7] 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.

[8] Best E., Devillers R., Koutny M. *Recursion and Petri nets. Acta Informaica* **37(11–12)**, p. 781–829, 2001, http://parsys.informatik.uni-oldenburg.de/~best/publications/bdk-recursion.ps.gz.

[9] Best E., Devillers R., Koutny M. *The box algebra = Petri nets + process expressions. Information and Computation* **178**, p. 44–100, 2002.

[10] 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.

[11] Buchholz P., Kemper P. *Quantifying the dynamic behavior of process algebras. Lecture Notes in Computer Science* **2165**, p. 184–199, 2001.

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

[13] Best E., Koutny M. *Solving recursive net equations. Lecture Notes in Computer Science* **944**, p. 605–623, 1995.

[14] Best E., Linde-Göers H.G. *Compositional process semantics of Petri boxes. Lecture Notes in Computer Science* **802**, p. 250–270, 1993.

[15] 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.

[16] Buchholz P., Tarasyuk I.V. *Net and algebraic approaches to probablistic modeling. Joint Novosibirsk Computing Center and Institute of Informatics Systems Bulletin, Series Computer Science* **15**, p. 31–64, Novosibirsk, 2001, http://www.iis.nsk.su/persons/itar/spnpancc.pdf.

[17] 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.

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

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

[20] Devillers R. *Construction of S-invariants and S-components for refined Petri boxes. Lecture Notes in Computer Science* **691**, p. 242–261, 1993.

[21] Devillers R. *S-invariant analysis of general recursive Petri boxes. Acta Informatica* **32(4)**, p. 313–345, 1995.

[22] Devillers R. *Petri boxes and finite processes. Lecture Notes in Computer Science* **1119**, p. 465–480, 1996.

[23] Donatelli S., Ribaudo M., Hillston J. *A comparison of perfomance 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.

[24] Esparza J., Bruns G. *Trapping mutual execution in the box calculus. Theoretical Computer Science* **153**, p. 95–128, 1996.

[25] de Frutos D.E., Marroquín O.A. *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`.

[26] Florin G., Natkin S. *Les reseaux de Petri stochastiques. Technique et Science Informatique* **4(1)**, 1985.

[27] 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.

[28] Hansson H. *Time and probability in formal design of distributed systems.* In: *Real-Time Safety Critical Systems*, Volume **1**, Elsevier, The Netherlands, 1994.

[29] Hillston J. *A compositional approach to performance modelling.* Cambridge University Press, Great Britain, 1996.

[30] Hesketh M., Koutny M. *An axiomatization of duplication equivalence in the Petri box calculus. Lecture Notes in Computer Science* **1420**, p. 165–184, 1998.

[31] 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.

[32] 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.

[33] 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`.

[34] Kotov V.E., Cherkasova L.A. *On structural properties of generalized processes. Lecture Notes in Computer Science* **188**, p. 288–306, 1985.

[35] Koutny M., Esparza J., Best E. *Operational semantics for the Petri box calculus. Lecture Notes in Computer Science* **836**, p. 210–225, 1994.

[36] Kotov V.E. *An algebra for parallelism based on Petri nets. Lecture Notes in Computer Science* **64**, p. 39–55, 1978.

[37] Koutny M. *Partial order semantics of box expressions. Lecture Notes in Computer Science* **815**, p. 318–337, 1994.

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

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

[40] Larsen K.G., Skou A. *Compositional verification of probabilistic processes. Lecture Notes in Computer Science* **630**, p. 456–471, 1992.

[41] Macià H.S. *sPBC: Una extensión Markoviana del Petri box calculus. Ph.D. thesis*, 249 p., Departamento de Informática, Universidad de Castilla-La Mancha, December 2003 (in Spanish), `http://www.info-ab.uclm.es/fmc/publications/2003/sPBCthesis03.pdf`.

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

[43] Marroquín O.A., de Frutos D.E. *Extending the Petri box calculus with time. Lecture Notes in Computer Science* **2075**, p. 303–322, 2001.

[44] Milner R.A.J. *Communication and concurrency.* Prentice-Hall International, New York, USA, 1989.

[45] Molloy M. *Performance analysis using stochastic Petri nets. IEEE Transactions on Software Engineering* **31(9)**, p. 913–917, 1982.

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

[47] Macià H.S., Valero V.R., Cuartero F.G. *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/descargas/tecnicalreports/DIAB-02-01-31/reportcong.pdf`.

[48] Macià H.S., Valero V.R., Cuartero F.G. *Defining equivalence relations in sPBC. Proceedings of* 1^{st} *International Conference on the Principles of Software Engineering*, 10 p., Buenos Aires, Argentina, November 2004, `http://www.info-ab.uclm.es/fmc/publications/2004/prise04.pdf`.

[49] Macià H.S., Valero V.R., Cazorlo D.L., Cuartero F.G. *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`.

[50] Macià H.S., Valero V.R., Cazorlo D.L., Cuartero F.G. *Introducing the iteration in sPBC. Proceedings of the* 24^{th} *International Conference on Formal Techniques for Networked and Distributed Systems (FORTE'04)*, Madrid, Spain, *Lecture Notes in Computer Science* **3235**, p. 292–308, October 2004, `http://www.info-ab.uclm.es/fmc/publications/2004/forte04.pdf`.

[51] Macià H.S., Valero V.R., Cuartero F.G., de Frutos D.E. *A congruence relation for sPBC.* 62 p., Department of Computer Science, University of Castilla-La Mancha, Albacete, Spain, June 2004 (submitted).

[52] Macià H.S., Valero V.R., Cuartero F.G., Pelayo F.L. *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/descargas/tecnicalreports/DIAB-02-01-26/newproposalsysPBC.ps`.

[53] Macià H.S., Valero V.R., Cuartero F.G., Pelayo F.L. *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.

[54] Macià H.S., Valero V.R., Cuartero F.G., Pelayo F.L. *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/fmc/publications/2003/acsd03.pdf`.

[55] Macià H.S., Valero V.R., Cuartero F.G., Ruiz M.C.D. *sPBC with immediate multiactions.* 27 p., Department of Computer Science, University of Castilla-La Mancha, Albacete, Spain, July 2005 (work in progress).

[56] Macià H.S., Valero V.R., de Frutos D.E. *sPBC: a Markovian extension of finite Petri box calculus. Proceedings of* 9^{th} *IEEE International Workshop 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/fmc/publications/2001/pnpm01.ps`.

[57] Macià H.S., Valero V.R., de Frutos D.E. *sPBC: a Markovian extension of finite PBC. Actas de IX Jornadas de Concurrencia - 01 (JC'01)*, 14 p., Sitges, Spain, June 2001,
`http://www.info-ab.uclm.es/fmc/publications/2001/mvfjc01.ps`.

[58] Macià H.S., Valero V.R., de Frutos D.E., Cuartero F.G. *Extending PBC with Markovian multiactions. Proceedings of XXVII Conferencia Latinoamericana de Informática - 01 (CLEI'01)*, 12 p., Mérida, Venezuela, September 2001, `http://www.info-ab.uclm.es/fmc/publications/2001/clei01.ps`.

[59] Núñez M., de Frutos D.E., Llana L. *Acceptance trees for probabilistic processes. Lecture Notes in Computer Science* **962**, p. 249–263, 1995.

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

[61] Núñez M. *An axiomatization of pobabilistic testing. Lecture Notes in Computer Science* **1601**, p. 130–150, 1995.

[62] 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.

[63] 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 (ISSN 0361-7688), 1998, `http://www.maik.rssi.ru/journals/procom.htm`.

[64] Tarasyuk I.V. *τ-equivalences and refinement for Petri nets based design. Technische Berichte* **TUD-FI00-11**, 41 p., Fakultät Informatik, Technische Universität Dresden, Germany, November 2000, `ftp://ftp.inf.tu-dresden.de/pub/berichte/tud00-11.ps.gz`.

[65] 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://www.iis.nsk.su/persons/itar/prlogncc.pdf`.

[66] Tarasyuk I.V. *Stochastic Petri nets: a formalism for modeling and performance analysis of computing processes. System Informatics* **9**, p. 135–194, Publishing House of SB RAS, Novosibirsk, 2004 (in Russian), `http://www.iis.nsk.su/persons/itar/spnsi.pdf`.

[67] Tarasyuk I.V., Buchholz P. *Equivalences for stochastic Petri nets and stochastic process algebras.* 35 p., 2005 (in Russian, submitted).

A Proof of Proposition 5.2

It is enough to prove it 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 $\mathcal{R} : G \underline{\leftrightarrow}_{sp} G'$ and $(s_1, s_2) \in \mathcal{R}$. By the definition of $\underline{\leftrightarrow}_{sp}$, we have $\forall A \in \mathbb{N}_f^{\mathcal{L}}\ \forall \widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$

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

Let $\mathcal{H} = [s_1]_{\mathcal{R}} = [s_2]_{\mathcal{R}}$. Then we can rewrite the above identity as

$$\mathcal{H} \xrightarrow{A}_{\mathcal{Q}} \widetilde{\mathcal{H}},$$

since for all states from the equivalence class $\mathcal{H}$ their probabilities of moving into $\widetilde{\mathcal{H}}$ as a result of execution of the multiset of multiactions A coincide (they are equal to $\mathcal{Q}$).

Note also that, starting from states of $TS^*(G)$ ($TS^*(G')$) to some set of states $\mathcal{H} \subseteq (DR(G) \cup DR(G'))$, we can reach only states of the same transition system, since the transition systems of two dynamic expressions do not communicate.

Let $(A_1 \cdots A_n, \mathcal{P}) \in StepProbTraces(G)$. Taking into account the previous notes and $\mathcal{R} : G \underline{\leftrightarrow}_{sp} G'$, we have

$$s_G \xrightarrow{A_1}_{\mathcal{Q}_1} \mathcal{H}_1 \cap DR(G) \xrightarrow{A_2}_{\mathcal{Q}_2} \cdots \xrightarrow{A_n}_{\mathcal{Q}_n} \mathcal{H}_n \cap DR(G) \ \Leftrightarrow\ s_{G'} \xrightarrow{A_1}_{\mathcal{Q}_1} \mathcal{H}_1 \cap DR(G') \xrightarrow{A_2}_{\mathcal{Q}_2} \cdots \xrightarrow{A_n}_{\mathcal{Q}_n} \mathcal{H}_n \cap DR(G').$$

Now we intend to prove that the sum of probabilities of all the paths going through the states from $\mathcal{H}_1 \cap DR(G), \ldots, \mathcal{H}_n \cap DR(G)$ coincides with the product of $\mathcal{Q}_1, \ldots, \mathcal{Q}_n$, which is essentially the probability of the "composite" path going through the equivalence classes $\mathcal{H}_1, \ldots, \mathcal{H}_n$ in $TS^*(G)$.

Lemma A.1 *For a dynamic expression G and all n $(1 \le n \le |DR(G)/_{\mathcal{R}}|)$ the following holds:*

$$\prod_{i=1}^{n} \mathcal{Q}_i = \sum_{\{\Gamma_1,\ldots,\Gamma_n \mid s_G \xrightarrow{\Gamma_1}_{\mathcal{P}_1} s_1 \xrightarrow{\Gamma_2}_{\mathcal{P}_2} \cdots \xrightarrow{\Gamma_n}_{\mathcal{P}_n} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^{n} \mathcal{P}_i.$$

Proof. (of the lemma) We shall prove by induction on n.

- $n = 1$

  We should prove that

$$\mathcal{Q}_1 = \sum_{\{\Gamma_1 \mid s_G \xrightarrow{\Gamma_1}_{\mathcal{P}_1} s_1,\ \mathcal{L}(\Gamma_1)=A_1,\ s_1 \in \mathcal{H}_1\}} \mathcal{P}_1.$$

  This follows from the definition of the transition relation between a state and a set of states.

- $n \rightarrow n+1$

  By induction hypothesis, we have the equality

$$\prod_{i=1}^{n} \mathcal{Q}_i = \sum_{\{\Gamma_1,\dots,\Gamma_n \mid s_G \xrightarrow{\Gamma_1}_{\mathcal{P}_1} s_1 \xrightarrow{\Gamma_2}_{\mathcal{P}_2} \cdots \xrightarrow{\Gamma_n}_{\mathcal{P}_n} s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i \in \mathcal{H}_i\ (1\le i\le n)\}} \prod_{i=1}^{n} \mathcal{P}_i.$$

  In addition, we have

$$\mathcal{Q}_{n+1} = \sum_{\{\Gamma_{n+1} \mid s_n \xrightarrow{\Gamma_{n+1}}_{\mathcal{P}_{n+1}} s_{n+1},\ \mathcal{L}(\Gamma_{n+1})=A_{n+1},\ s_{n+1} \in \mathcal{H}_{n+1}\}} \mathcal{P}_{n+1},$$

  again by the definition of the transition relation between a state and a set of states. Let us note that the sum above does not depend on particular $s_n \in \mathcal{H}_n$, i.e., it is the *same for all paths* of $TS^*(G)$ starting at s_G and going through $\mathcal{H}_1, \dots, \mathcal{H}_n$.

  As a result of multiplying the left and the right part of the two above equalities, we obtain

$$\prod_{i=1}^{n} \mathcal{Q}_i \cdot \mathcal{Q}_{n+1} = \left(\sum_{\{\Gamma_1,\dots,\Gamma_n \mid s_G \xrightarrow{\Gamma_1}_{\mathcal{P}_1} s_1 \xrightarrow{\Gamma_2}_{\mathcal{P}_2} \cdots \xrightarrow{\Gamma_n}_{\mathcal{P}_n} s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i \in \mathcal{H}_i\ (1\le i\le n)\}} \prod_{i=1}^{n} \mathcal{P}_i \right) \cdot$$
$$\sum_{\{\Gamma_{n+1} \mid s_n \xrightarrow{\Gamma_{n+1}}_{\mathcal{P}_{n+1}} s_{n+1},\ \mathcal{L}(\Gamma_{n+1})=A_{n+1},\ s_{n+1} \in \mathcal{H}_{n+1}\}} \mathcal{P}_{n+1}.$$

  By distributivity and taking into account the note above on independence of the sum of current probabilities from the concrete state s_n, we conclude that

$$\prod_{i=1}^{n+1} \mathcal{Q}_i = \sum_{\{\Gamma_1,\dots,\Gamma_{n+1} \mid s_G \xrightarrow{\Gamma_1}_{\mathcal{P}_1} s_1 \xrightarrow{\Gamma_2}_{\mathcal{P}_2} \cdots \xrightarrow{\Gamma_{n+1}}_{\mathcal{P}_{n+1}} s_{n+1},\ \mathcal{L}(\Gamma_i)=A_i,\ s_i \in \mathcal{H}_i\ (1\le i\le n+1)\}} \prod_{i=1}^{n+1} \mathcal{P}_i.$$

This ends the proof of the lemma. □ (the lemma)

Note that the result of this lemma can also be applied to G'.

Now we only need to see that summation over *all equivalence classes* is the same as summation over *all states*, hence, over *all multisets of activities*, since their executions result the states, i.e.,

$$\sum_{\{\Gamma_1,\dots,\Gamma_n \mid s_G \xrightarrow{\Gamma_1}_{\mathcal{P}_1} s_1 \xrightarrow{\Gamma_2}_{\mathcal{P}_2} \cdots \xrightarrow{\Gamma_n}_{\mathcal{P}_n} s_n,\ \mathcal{L}(\Gamma_i)=A_i\ (1\le i\le n)\}} \prod_{i=1}^{n} \mathcal{P}_i =$$
$$\sum_{\{\mathcal{H}_1,\dots,\mathcal{H}_n \mid s_G \xrightarrow{A_1}_{\mathcal{Q}_1} \mathcal{H}_1 \cap DR(G) \xrightarrow{A_2}_{\mathcal{Q}_2} \cdots \xrightarrow{A_n}_{\mathcal{Q}_n} \mathcal{H}_n \cap DR(G)\}} \prod_{i=1}^{n} \mathcal{Q}_i =$$
$$\sum_{\{\mathcal{H}_1,\dots,\mathcal{H}_n \mid s_{G'} \xrightarrow{A_1}_{\mathcal{Q}_1} \mathcal{H}_1 \cap DR(G') \xrightarrow{A_2}_{\mathcal{Q}_2} \cdots \xrightarrow{A_n}_{\mathcal{Q}_n} \mathcal{H}_n \cap DR(G')\}} \prod_{i=1}^{n} \mathcal{Q}_i =$$

$$\sum_{\{\Gamma'_1,\ldots,\Gamma'_n \mid s_{G'} \xrightarrow{\Gamma'_1}_{\mathcal{P}'_1} s'_1 \xrightarrow{\Gamma'_2}_{\mathcal{P}'_2} \cdots \xrightarrow{\Gamma'_n}_{\mathcal{P}'_n} s'_n,\ \mathcal{L}(\Gamma'_i)=A_i\ (1\leq i\leq n)\}} \prod_{i=1}^{n} \mathcal{P}'_i.$$

Hence, $(A_1 \cdots A_n, \mathcal{P}) \in StepProbTraces(G')$, and we have $StepProbTraces(G) \subseteq StepProbTraces(G')$. The reverse inclusion is proved by symmetry. □