Fundamenta Informaticae 76 (2007) 189–219
IOS Press

Stochastic Petri box calculus with discrete time

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

Abstract. In the last decades, a number of stochastic enrichments of process algebras was constructed to allow one for specification of stochastic processes within the well-developed framework of algebraic calculi. In [40], a continuous time stochastic extension of finite Petri box calculus (PBC) was proposed called $sPBC$. The 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. A step operational semantics is defined in terms of labeled transition systems based on action and inaction rules. A denotational semantics is defined in terms of a subclass of labeled discrete time stochastic Petri nets (LDTSPNs) called discrete time stochastic Petri boxes (dts-boxes). A consistency of both semantics is demonstrated. In addition, we define a variety of probabilistic equivalences that allow one to identify stochastic processes with similar behaviour which are differentiated by too strict notion of the semantic equivalence. The interrelations of all the introduced equivalences are investigated.

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 [43, 44, 17]. 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

Address for correspondence: A.P. Ershov Institute of Informatics Systems, Siberian Branch of the Russian Academy of Sciences, 6, Acad. Lavrentiev ave., 630090 Novosibirsk, Russia

Markov chain generated and analyzed by well-developed algorithms and methods. Firing probabilities distributed along continuous or discrete time scale are associated with transitions of an SPN. Thus, there exist SPNs with continuous 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 an interleaving one. In this semantics, parallel computations are modeled by all possible execution sequences of their components. Transitions of DTSPNs fire concurrently in steps at discrete time moments. Hence, this model has a step semantics. In this semantics, parallel computations are modeled by sequences of concurrent occurrences (steps) of their components. In [10, 11], a labeling for transitions of CTSPNs with action names was proposed. The labeling allows SPNs to model processes with functionally similar components: the transitions corresponding to the similar components are labeled by the same action. Moreover, one can compare labeled SPNs by different behavioural equivalences, and this makes possible to check stochastic processes specified by labeled SPNs for functional similarity. Therefore, one can compare both functional and performance properties, and labeled SPNs turn into a formalism for quantitative and qualitative analysis.

Algebraic calculi occupy a special place among formal models for specification of concurrent systems and analysis of their behavioral properties. In such process algebras (PAs), a system or a process is specified by an algebraic formula. 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) [42]. Process algebras have been acknowledged to be very suitable formalism to operate with real time and stochastic systems as well. In the last years, stochastic extensions of PAs called stochastic process algebras (SPAs) became very popular as a modeling framework. SPAs do not just specify actions that can happen (qualitative features) as usual process algebras, but they associate some quantitative parameters with actions (quantitative characteristics). The papers [20, 9, 21, 16, 47, 12] 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 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 based on interleaving. As a semantic domain, the interleaving formalism of transition systems is used. For example, an extension of CCS with probabilities and time called $TPCCS$ was defined in [19]. 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 [22]. The most popular SPAs proposed so far are $PEPA$ [21], $TIPP$ [20] and $EMPA$ [2]. It is worth to mention the stochastic process calculus PPA constructed in [46, 45] as well. Therefore, an investigation of a stochastic extension for more expressive and powerful algebraic calculi is an important issue. At present, the development of step or "true concurrency" (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 [14]. 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 [42] and AFP_0 [23]. PBC was introduced more than 10 years ago [4], and it was well explored since that time [3, 25, 8, 15, 5, 26, 6, 7]. 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. The calculus PBC has a step operational semantics in terms of labeled transition systems based on structural operational semantics (SOS) rules. A pomset operational semantics of PBC was defined in [25] 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.

A stochastic extension of PBC called stochastic Petri box calculus ($sPBC$) was proposed in [40, 39, 41, 31, 35, 34, 36, 29]. 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. A denotational semantics was defined in terms of a subclass of labeled continuous time stochastic Petri nets (CTSPNs) called stochastic Petri boxes (s-boxes). The 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 [38, 30]. In the papers [32, 33], a number of new equivalence relations were proposed for regular terms of $sPBC$ to choose later a suitable candidate for a congruence. In [37], the special multiactions with zero time delay were added to $sPBC$. A 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). 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 a step operational semantics.

A notion of equivalence is very important in formal theory of computing processes and systems. Behavioural equivalences are applied during verification stage both to compare behaviour of systems and reduce their structure. At present time, there exists a great diversity of different equivalence notions for concurrent systems, and their interrelations were well explored in the literature. The most popular and widely used one is bisimulation. Unfortunately, the mentioned behavioural equivalences take into account only functional (qualitative) but not performance (quantitative) aspects of system behaviour. Additionally, the equivalences are often interleaving ones, and they do not respect concurrency. SPAs inherited from 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 [12], a notion of interleaving stochastic bisimulation equivalence for process terms was introduced. 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.

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 [13], labeled weighted discrete time SPNs (LWDTSPNs) were proposed that is a modification of DTSPNs by transition labeling and weights. Transitions of LWDTSPNs 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 LWDTSPNs was demonstrated. In [14], a stochastic algebra of finite nondeterministic processes $StAFP_0$ was proposed with semantics in terms of a subclass of LWDTSPNs and LDTSPNs called stochastic acyclic nets (SANs). The calculus defined is a stochastic extension of algebra AFP_0 introduced in [24]. The 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 concurrency" model. Usually, transition systems are used for this purpose, but they are not able to respect concurrency completely. An axiomatization for the semantic equivalence of $StAFP_0$ was proposed. It was proved that any algebraic formula could be reduced to the "fully stratified" one with the use of the axiom system. This simplifies semantic comparison of formulas.

In this paper, we propose a discrete time stochastic extension of finite PBC called $dtsPBC$. The work consists of the following stages. First, we present the syntax of $dtsPBC$. Each multiaction of the initial calculus PBC is associated with a probability. Such a pair is called stochastic multiaction or activity. Second, we propose semantics of $dtsPBC$. A step operational semantics is constructed in terms of labeled transition systems based on action and inaction rules. The difficulty 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. The denotational semantics is defined in terms of a subclass of labeled DTSPNs (LDTSPNs) called discrete time stochastic Petri boxes (dts-boxes). A consistency of operational and denotational semantics is proved. In the last part, 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. The earlier report on the results presented here is [49].

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 [4]. 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 $\forall x \in X\ M(x) \leq 1$, 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 $\forall x \in X\ M(x) \leq M'(x)$. 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 $\widehat{Act} = \{\hat{a}, \hat{b}, \ldots\}$ is the set of *conjunctive actions (conjugates)* such that $a \neq \hat{a}$ 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 that contains no visible action names. The *alphabet* of $\alpha \in \mathcal{L}$ is defined as $\mathcal{A}(\alpha) = \{x \in \mathcal{A} \mid \alpha(x) > 0\}$.

An *activity (stochastic multiaction)* is a pair (α, ρ), where $\alpha \in \mathcal{L}$ and $\rho \in (0; 1)$ is the probability of the multiaction α. The multiaction probabilities are used to calculate probabilities of state changes (steps) at discrete time moments. The multiaction probabilities are required not to be equal to 1, since otherwise, the multiactions with probability 1 always happen in a step, and all other with the less probabilities do not. In this case, technical difficulties appear with conflicts resolving, see [44]. 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* $[f]$, *restriction* rs and *synchronization* sy.

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 the structure of a system. As we shall see, they agree 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 the states of a system. As we shall see, they agree to marked SPNs.

Definition 2.3. Let $(\alpha,\rho) \in \mathcal{SL},\ a \in Act$ and $E \in StatExpr$. 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 a step operational semantics in terms of labeled transition systems.

3.1. Inaction rules

First, we define inaction rules for overlined and underlined static expressions. Let $E, F \in StatExpr$ and $a \in Act$.

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

Second, we propose inaction rules for arbitrary dynamic expressions. Let $E \in StatExpr,\ G, H, \widetilde{G}, \widetilde{H} \in DynExpr$ and $a \in Act$.

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

Note that the rule $G \stackrel{\emptyset}{\rightarrow} G$ 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.

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

Definition 3.1. Let $\simeq\ =\ (\stackrel{\emptyset}{\rightarrow} \cup \stackrel{\emptyset}{\leftarrow})^*$ 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 from each other by applying inaction rules.

3.2. Action rules

Now we propose action rules which 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 $\mathcal{A}(\Gamma) = \cup_{(\alpha,\rho)\in\Gamma}\mathcal{A}(\alpha)$.

$$\overline{(\alpha,\rho)} \stackrel{\{(\alpha,\rho)\}}{\longrightarrow} \underline{(\alpha,\rho)} \qquad \frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G}}{G;E \stackrel{\Gamma}{\rightarrow} \widetilde{G};E} \qquad \frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G}}{E;G \stackrel{\Gamma}{\rightarrow} E;\widetilde{G}} \qquad \frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G}}{G[]E \stackrel{\Gamma}{\rightarrow} \widetilde{G}[]E} \qquad \frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G}}{E[]G \stackrel{\Gamma}{\rightarrow} E[]\widetilde{G}} \qquad \frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G}}{G\|H \stackrel{\Gamma}{\rightarrow} \widetilde{G}\|H}$$

$$\frac{H \stackrel{\Gamma}{\rightarrow} \widetilde{H}}{G\|H \stackrel{\Gamma}{\rightarrow} G\|\widetilde{H}} \qquad \frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G},\ H \stackrel{\Delta}{\rightarrow} \widetilde{H}}{G\|H \stackrel{\Gamma+\Delta}{\longrightarrow} \widetilde{G}\|\widetilde{H}} \qquad \frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G}}{G[f] \stackrel{f(\Gamma)}{\longrightarrow} \widetilde{G}[f]} \qquad \frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G},\ a,\hat{a} \not\in \mathcal{A}(\Gamma)}{G \text{ rs } a \stackrel{\Gamma}{\rightarrow} \widetilde{G} \text{ rs } a} \qquad \frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G}}{G \text{ sy } a \stackrel{\Gamma}{\rightarrow} \widetilde{G} \text{ sy } a}$$

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

Note that in the last rule we multiply the 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 the 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. The new activities generated from the synchronization will be annotated with the concatenation of the numbering of the activities they come from. Such new activities will be considered up to the permutation of their numbering resulting from the applications of the second rule for synchronization. After such an enumeration the multisets of activities over arrows in the action rules will be proper sets.

Definition 3.2. Let G be a dynamic expression. Then $[G]_\simeq = \{H \mid G \simeq H\}$ is the equivalence class of G with respect to isomorphism. The *derivation set* of G, denoted by $DR(G)$, is the minimal set such that

- $[G]_\simeq \in DR(G)$;
- if $[H]_\simeq \in DR(G)$ and $\exists\Gamma\ H \stackrel{\Gamma}{\rightarrow} \widetilde{H}$ 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 $Exec(H) = \{\Gamma \mid \exists J \in [H]_\simeq\ \exists\widetilde{J}\ J \stackrel{\Gamma}{\rightarrow} \widetilde{J}\}$.

Let $\Gamma \in Exec(H)$. The probability that the activities from Γ *try to happen* in H is

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

When $Exec(H) = \emptyset$, we define $PF(\emptyset, H) = 1$, since we stay in H in this case.

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

The 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 tries to happen *normalized* by the probability to occur for *any* multiset executable from H. The denominator of the fraction above is a summation since it reflects the probability of the event union.

The probability that the execution of *any* activities changes H to $\widetilde{H}$ 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 the probability for *any* multiset of activities to change H to $\widetilde{H}$, we use summation in the definition.

Definition 3.3. Let G be a dynamic expression. The *(labeled probabilistic) transition system* of G is a quadruple $TS(G) = (S_G, L_G, \rightarrow_G, s_G)$, where

- the set of *states* is $S_G = DR(G)$;
- the set of *labels* is $L_G \subseteq I\!N_f^{\mathcal{SL}} \times (0; 1]$;
- the set of *transitions* is $\rightarrow_G = \{([H]_\simeq, (\Gamma, PT(\Gamma, H)), [\widetilde{H}]_\simeq) \mid [H]_\simeq \in DR(G),\ H \xrightarrow{\Gamma} \widetilde{H}\}$;
- 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 (one-step) 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 $[G]_\simeq$. A transition $(s, (\Gamma, \mathcal{P}), \tilde{s}) \in \rightarrow_G$ will be written as $s \xrightarrow{\Gamma}_{\mathcal{P}} \tilde{s}$. It is interpreted as follows: the probability to change the state s to $\tilde{s}$ as a result of executing Γ is $\mathcal{P}$. The step probabilities belong to the interval $(0; 1]$. The value 1 is the case when we cannot leave a state, and thus there exists the only transition from the state to itself.

We write $s \xrightarrow{\Gamma} \tilde{s}$ if $\exists\mathcal{P}\ s \xrightarrow{\Gamma}_{\mathcal{P}} \tilde{s}$. 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 empty set, and its execution does not change isomorphism 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. It follows from the definition of $PF(\emptyset, H)$ and the fact that multiaction probabilities cannot be equal to 1 as they belong to the interval $(0; 1)$.

Definition 3.4. Let G, G' be dynamic expressions and $TS(G) = (S_G, L_G, \rightarrow_G, s_G)$, $TS(G') = (S_{G'}, L_{G'}, \rightarrow_{G'}, s_{G'})$ be their transition systems. A mapping $\beta : S_G \rightarrow S_{G'}$ is an *isomorphism* between $TS(G)$ and $TS(G')$, denoted by $\beta : TS(G) \simeq TS(G')$, if β is a bijection such that $\beta(s_G) = s_{G'}$ and $\forall s, \tilde{s} \in S_G\ \forall\Gamma\ s \xrightarrow{\Gamma}_{\mathcal{P}} \tilde{s} \ \Leftrightarrow\ \beta(s) \xrightarrow{\Gamma}_{\mathcal{P}} \beta(\tilde{s})$. Two transition systems $TS(G)$ and $TS(G')$ are *isomorphic*, denoted by $TS(G) \simeq TS(G')$, if $\exists\beta : TS(G) \simeq TS(G')$.

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

Definition 3.5. Two dynamic expressions G and G' are *isomorphic with respect to transition systems*, denoted by $G =_{ts} G'$, if $TS(G) \simeq TS(G')$.

Definition 3.6. 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 we have $PM(H, \widetilde{H}) = \sum_{\{\Gamma \mid [H]_{\simeq} \xrightarrow{\Gamma}_{\mathcal{P}} [\widetilde{H}]_{\simeq}\}} \mathcal{P}$, 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 $\tilde{s}$.

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

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

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.

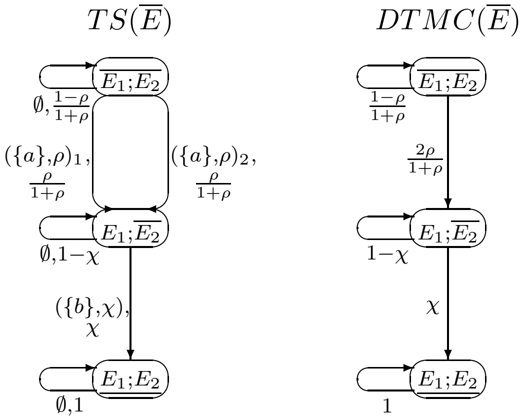

Figure 1. The transition system and the underlying DTMC of $\overline{E}$ for $E = ((\{a\},\rho)_1[](\{a\},\rho)_2);(\{b\},\chi)$

4.1. Labeled DTSPNs

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

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

- P_N and T_N are finite sets of *places* and *transitions*, respectively, such that $P_N \cup T_N \neq \emptyset$ and $P_N \cap T_N = \emptyset$;
- $W_N : (P_N \times T_N) \cup (T_N \times P_N) \to I\!N$ is a function describing the *weights of arcs* between places and transitions;
- $\Omega_N : T_N \to (0;1)$ is the *transition probability* function associating transitions with probabilities;
- $L_N : T_N \to Act_\tau$ 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 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 arc weights are depicted near them. 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 such that each of them is enabled in a marking* M. A set of transitions

$U \subseteq Ena(M)$ is enabled in a marking M if ${}^\bullet U \subseteq M$. Firings of transitions are atomic operations, and transitions may fire concurrently in steps. We assume that all transitions participating in a step should differ, hence, only sets (not multisets) of transitions may fire. Thus, we do not allow self-concurrency, i.e., firing of transitions concurrently to themselves. This restriction is introduced because we would like to avoid technical difficulties while calculating probabilities for multisets of transitions as we shall see after the following formal definitions.

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

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

In the case $U = \emptyset$ we define

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

Thus, $PF(U, M)$ could be interpreted as a *joint* probability of independent events. Each such an event is interpreted as trying or not trying to fire of a particular transition from U. The multiplication in the definition is used because it reflects the probability of event intersection. When no transitions are enabled in M, we have $PF(\emptyset, M) = 1$, since we stay in M in this case.

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

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

In the case $U = \emptyset$ we have $M = \widetilde{M}$ and

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

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

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

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 $\exists U\ M \xrightarrow{U} \widetilde{M}$ then $\widetilde{M} \in RS(N)$.

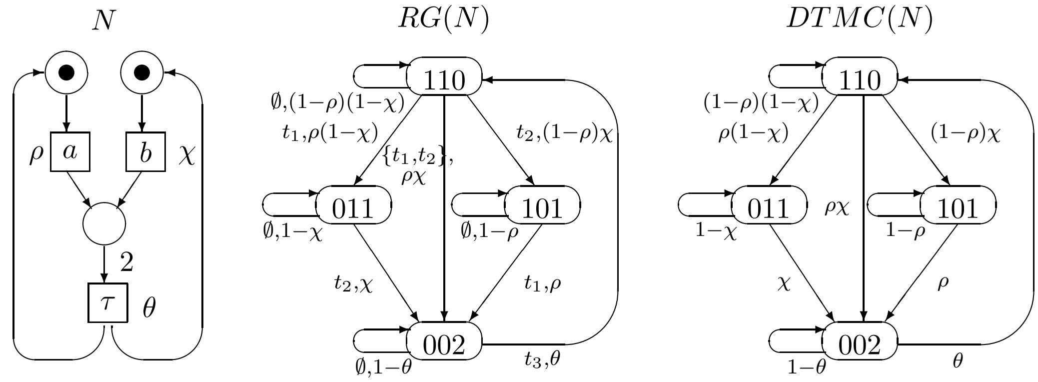

Figure 2. LDTSPN, its reachability graph and the underlying DTMC

- 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 $M \xrightarrow{U}_{\mathcal{P}} \widetilde{M}$.
- 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

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

Thus, $PM(M, \widetilde{M})$ is the probability for *any* transition set to change marking M to $\widetilde{M}$, hence we use summation in the definition.

Example 4.1. In Figure 2 an LDTSPN with two visible transitions t_1 (labeled by a), t_2 (labeled by b) and one invisible transition t_3 (labeled by τ) is depicted. Transition probabilities of N are denoted by $\rho = \Omega_N(t_1),\ \chi = \Omega_N(t_2),\ \theta = \Omega_N(t_3)$. In the figure one can see the reachability graph $RG(N)$ and the underlying DTMC $DTMC(N)$ as well. 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 is required to hold: ${}^\circ N \neq \emptyset \neq N^\circ,\ {}^\bullet({}^\circ N) = \emptyset = (N^\circ)^\bullet$.

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*. We shall use the following notation: $\overline{N} = (N, {}^\circ N)$ and $\underline{N} = (N, N^\circ)$. Note that a marked plain dts-box $(P_N, T_N, W_N, \Lambda_N, M_N)$ could be interpreted as the LDTSPN $(P_N, T_N, W_N, \Omega_N, L_N, M_N)$, 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 which label unobservable transitions of the corresponding dts-box. The behaviour of marked dts-boxes follows from the firing rule of LDTSPNs. A plain dts-box N is *safe*, if $\overline{N}$ is, i.e., $\forall M \in RS(\overline{N})\ M \subseteq P_N$. A 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 associates the numbers with transitions of a plain dts-box N in accordance with 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 numbering of the transitions it comes from with the resulting new transition. The transitions resulting from synchronization are considered up to the permutation of their numbering resulting from the applications of the second rule for synchronization to the corresponding expression.

The structure of the plain dts-box corresponding to a static expression is constructed as in PBC, see [8, 6]. I.e., we use simultaneous refinement and relabeling meta-operator (net refinement) in addition to the *operator dts-boxes* corresponding to the algebraic operations of $dtsPBC$ and featuring transformational transition relabelings. Thus, the resulting plain dts-boxes are safe and clean. In the definition of denotational semantics we shall use standard constructions used for PBC. For convenience, we only use slightly different notation: ϱ, Θ and u stand for ρ (relabeling), Ω (operator box) and v (transition name) from PBC setting, respectively.

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

- $\varrho_{id} = \{(\{(\alpha, \rho)\}, (\alpha, \rho) \mid (\alpha, \rho) \in \mathcal{SL}\}$ is the *identity* relabeling keeping the interface as it is;
- $\varrho_{[f]} = \{(\{(\alpha, \rho)\}, (f(\alpha), \rho) \mid (\alpha, \rho) \in \mathcal{SL}\}$;
- $\varrho_{\mathsf{rs}\ a} = \{(\{(\alpha, \rho)\}, (\alpha, \rho) \mid (\alpha, \rho) \in \mathcal{SL},\ a, \hat{a} \not\in \mathcal{A}(\alpha)\}$;
- $\varrho_{\mathsf{sy}\ a}$ 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 $(\Gamma + \Delta, \{(\alpha + \beta, \rho \cdot \chi)\} \in \varrho_{\mathsf{sy}\ a}$.

The plain and operator dts-boxes are presented in Figure 3. The symbol i is usually omitted.

Now we define the 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.

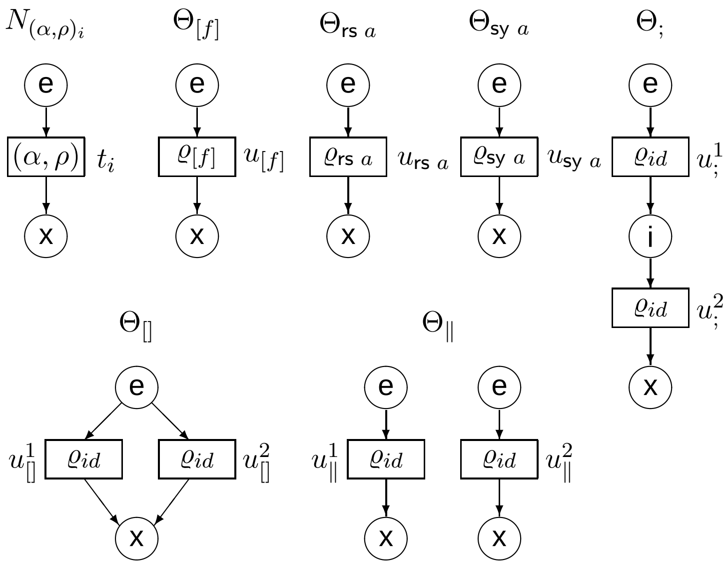

Figure 3. The plain and operator dts-boxes

- $Box_{dts}(E \circ F) = \Theta_\circ(Box_{dts}(E), Box_{dts}(F)),\ \circ \in \{;, [], \|\}$. 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 the labels of some multiactions by a bijection, we preserve the initial enumeration:

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

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

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

- $Box_{dts}(E \text{ sy } a) = \Theta_{\text{sy } a}(Box_{dts}(E))$. 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 $L(t) = \alpha + \beta$, 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 the same transition set in a different order, we only consider a single one of them in the plain dts-box.

Now we can formally define denotational semantics as a homomorphism.

Definition 4.4. Let $(\alpha, \rho) \in \mathcal{SL},\ a \in Act$ and $E, F \in StatExpr$. The *denotational semantics* of $dtsPBC$ 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. $Box_{dts}(E \circ a) = \Theta_{\circ a}(Box_{dts}(E)),\ \circ \in \{\mathsf{rs}, \mathsf{sy}\}$.

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 overlined or underlined static expressions or those without overlines and underlines, and the definition of dts-boxes is compositional.

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

Theorem 4.1. For any static expression E

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

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 the probabilities of the transitions belonging to the corresponding plain dts-box. Second, in both semantics, conflicts are resolved via the same probability functions. □

Proposition 4.1. For any static expression E

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

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

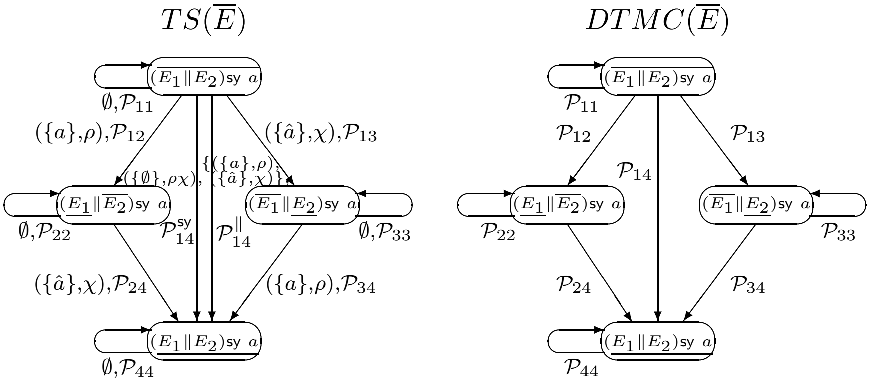

Figure 4. The transition system and the underlying DTMC of $\overline{E}$ for $E = ((\{a\},\rho)\|(\{\hat{a}\},\chi))\ \mathsf{sy}\ a$

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 $N = Box_{dts}(\overline{E})$, its reachability graph $RG(N)$ and the underlying DTMC $DTMC(N)$ are presented. It is easy to see that $TS(\overline{E})$ and $RG(N)$ are isomorphic as well as $DTMC(\overline{E})$ and $DTMC(N)$.

The 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 $\mathcal{N} = \frac{1}{1-\rho^2\chi-\rho\chi^2+\rho^2\chi^2}$.

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

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.

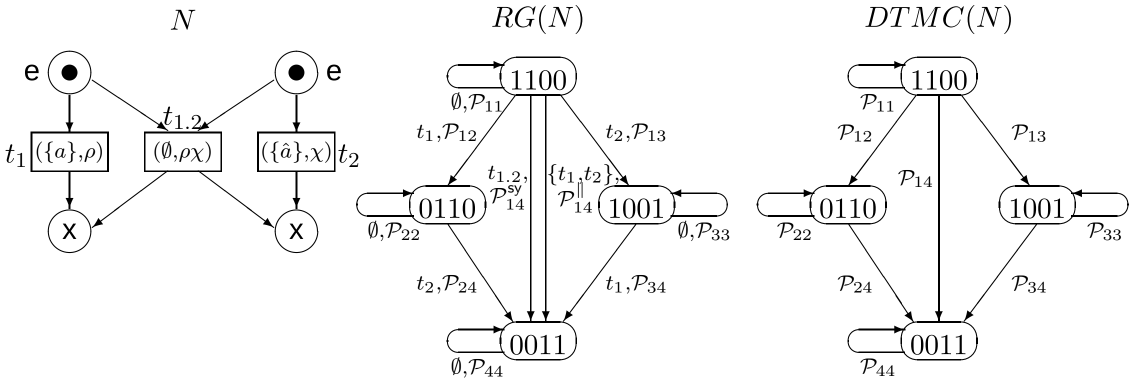

Figure 5. The marked dts-box $N = Box_{dts}(\overline{E})$ for $E = ((\{a\},\rho)\|(\{\hat{a}\},\chi))$ sy a, its reachability graph and the underlying DTMC

To 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 an 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 from empty loops both in the algebraic setting of $dtsPBC$ and in the net one of LDTSPNs.

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 the empty set and have non-zero probability. The *empty loop* $s \xrightarrow{\emptyset}_{\mathcal{P}} s$ 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 $(k \geq 1)$ *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

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

The value $k = 0$ in the summation above corresponds to the case when no empty loops occur. Note that $PT^*(\Gamma, H) \leq 1$, hence, it is really a probability, since $PT(\emptyset, H) + PT(\Gamma, H) \leq PT(\emptyset, H) + \sum_{\Delta \in Exec(H) \setminus \emptyset} PT(\Delta, H) = \sum_{\Delta \in Exec(H)} PT(\Delta, H) = 1$.

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

We decided to consider an empty loop followed by a non-empty step only 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 [18].

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

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

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

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

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

When concurrency aspects are not relevant, interleaving behaviour is 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 $s \stackrel{(\alpha,\rho)}{\twoheadrightarrow}_{\mathcal{Q}} \tilde{s}$ if $s \stackrel{(\alpha,\rho)}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s}$ and $\mathcal{Q} = \frac{\mathcal{P}}{\sum_{\{\{(\beta,\chi)\} \in Exec(H),\ \bar{s} \in DR(G) \mid s \stackrel{(\beta,\chi)}{\twoheadrightarrow}_{\overline{\mathcal{P}}} \bar{s}\}} \overline{\mathcal{P}}}$.

5.2. Empty loops in reachability graphs

Let N be an LDTSPN. Reachability graph $RG(N)$ can have loops going from a state to itself which are labeled by an emptyset and have non-zero probability. The *empty loop* $M \stackrel{\emptyset}{\rightarrow}_{\mathcal{P}} M$ 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.

Let N be an LDTSPN and $M \in RS(N)$. The *probability to stay in M due to k ($k \geq 1$) empty loops* is $(PT(\emptyset, M))^k$. The *probability to execute in M a non-empty transition set U after possible empty loops* is

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

The value $k = 0$ in the summation above corresponds to the case when no empty loops occur. Note that $PT^*(U,M) \leq 1$, hence, it is really a probability, since $PT(\emptyset,M) + PT(U,M) \leq PT(\emptyset,M) + \sum_{\{V|^\bullet V \subseteq M\}} PT(V,M) = 1$.

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 $M \stackrel{U}{\twoheadrightarrow}_{PT^*(U,M)} \widetilde{M}$, if $M \stackrel{U}{\rightarrow} \widetilde{M},\ U \neq \emptyset$.

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

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

We decided to consider an empty loop followed by a non-empty step only 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 [18].

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 $\exists U\ M \stackrel{U}{\twoheadrightarrow} \widetilde{M}$, where the transition probability is

$$PM^*(M,\widetilde{M}) = \sum_{\{U \in Ena(M) | M \stackrel{U}{\twoheadrightarrow} \widetilde{M}\}} PT^*(U,M).$$

When concurrency aspects are not relevant, interleaving behaviour is 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 $M \stackrel{t}{\rightarrowtail}_{\mathcal{Q}} \widetilde{M}$ if $M \stackrel{t}{\twoheadrightarrow}_{\mathcal{P}} \widetilde{M}$ and $\mathcal{Q} = \frac{\mathcal{P}}{\sum_{\{u \in Ena(M),\ \overline{M} \in RS(N) | M \stackrel{u}{\twoheadrightarrow}_{\overline{\mathcal{P}}} \overline{M}\}} \overline{\mathcal{P}}}$.

Theorem 5.1. For any static expression E

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

Proof:
As Theorem 4.1. □

Proposition 5.1. For any static expression E

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

Proof:
As Proposition 4.1. □

Note that Theorem 5.1 guarantees that the net versions of algebraic equivalences could be easily defined. For every equivalence on the 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 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 $\mathcal{N}^* = \frac{1}{\rho+\chi-2\rho^2\chi-2\rho\chi^2+2\rho^2\chi^2}$. 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 discriminating ones. In the trace semantics, the behavior of a system is associated with the set of all possible sequences of activities, i.e., protocols of work or computations. Thus, the points of choice of an external observer between several extensions of a particular computation are not taken into account.

Formal definitions of probabilistic trace relations resemble those of trace equivalences for standard Petri nets [48] 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

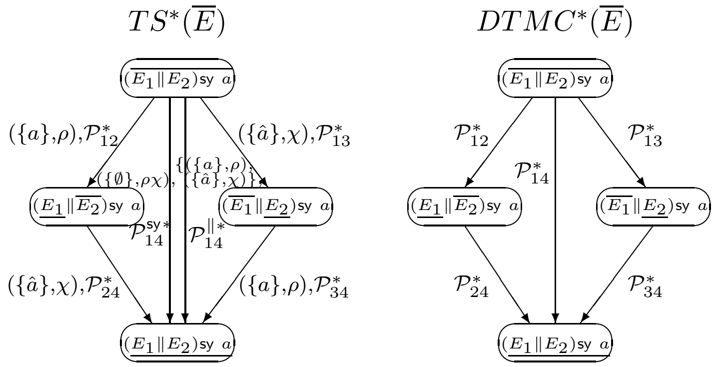

Figure 6. The transition system and the underlying DTMC without empty loops of $\overline{E}$ from Example 4.2

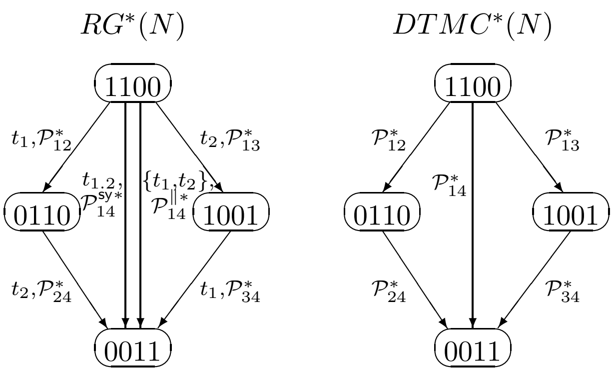

Figure 7. The reachability graph and the underlying DTMC without empty loops of N from Example 4.2

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.

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

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

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 $TS(G) = (S_G, L_G, \rightarrow_G, s_G)$ is a pair $(\Sigma, \mathcal{P})$, where $\Sigma = A_1 \cdots A_n \in (I\!N_f^{\mathcal{L}})^*$ and

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

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 [27, 28]. 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 $A \in \mathbb{N}_f^{\mathcal{L}}$ we write $s \xrightarrow{A}\!\!\!\!\rightarrow_{\mathcal{Q}} \mathcal{H}$ if

$$\mathcal{Q} = \sum_{\{\Gamma \mid s \xrightarrow{\Gamma}\!\!\!\!\rightarrow_{\mathcal{P}} \tilde{s},\ \mathcal{L}(\Gamma)=A,\ \tilde{s}\in\mathcal{H}\}} \mathcal{P}.$$

Thus, $\mathcal{Q}$ is the overall probability to come into the set of states $\mathcal{H}$ starting from s via steps with multiaction part A. The summation above reflects the probability of the event union.

We write $s \xrightarrow{A}\!\!\!\!\rightarrow \mathcal{H}$ if $\exists \mathcal{Q}\ s \xrightarrow{A}\!\!\!\!\rightarrow_{\mathcal{Q}} \mathcal{H}$. In a similar way, we define the notions $s \xrightharpoonup{\alpha}_{\mathcal{Q}} \mathcal{H}$ and $s \xrightharpoonup{\alpha} \mathcal{H}$ based on the interleaving transition relation.

For a set X, we denote its cartesian product $X \times X$ by X^2. Let $\mathcal{E} \subseteq X^2$ be an equivalence relation on X. Then an *equivalence class* (with respect to $\mathcal{E}$) of $x \in X$ is $[x]_{\mathcal{E}} = \{y \in X \mid (x, y) \in \mathcal{E}\}$. The equivalence $\mathcal{E}$ partitions X into the *set of equivalence classes* $X/_{\mathcal{E}} = \{[x]_{\mathcal{E}} \mid x \in X\}$.

Definition 5.8. Let G be a dynamic expression and $TS(G) = (S_G, L_G, \rightarrow_G, s_G)$ 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 $\star = i$;
- $\forall x \in \mathbb{N}_f^{\mathcal{L}}$ 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 $TS(G) = (S_G, L_G, \rightarrow_G, s_G)$, $TS(G') = (S_{G'}, L_{G'}, \rightarrow_{G'}, s_{G'})$ 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 $TS(G) = (S_G, L_G, \rightarrow_G, s_G)$, $TS(G') = (S_{G'}, L_{G'}, \rightarrow_{G'}, s_{G'})$ be their transition systems. A mapping $\beta : S_G \rightarrow S_{G'}$ is a *stochastic isomorphism* between G and G', denoted by $\beta : G =_{sto} G'$, if

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

2. $\forall s, \tilde{s} \in S_G$ if $s \xrightarrow{\Gamma}\!\!\!\!\rightarrow_{\mathcal{P}} \tilde{s}$ then $\exists \Gamma', \mathcal{P}'$ such that $\beta(s) \xrightarrow{\Gamma'}\!\!\!\!\rightarrow_{\mathcal{P}'} \beta(\tilde{s}),\ \mathcal{L}(\Gamma) = \mathcal{L}(\Gamma')$ and

$$\sum_{\{\Delta \mid s \xrightarrow{\Delta}\!\!\!\!\rightarrow_{\mathcal{Q}} \tilde{s},\ \mathcal{L}(\Gamma)=\mathcal{L}(\Delta)\}} \mathcal{Q} = \sum_{\{\Delta' \mid \beta(s) \xrightarrow{\Delta'}\!\!\!\!\rightarrow_{\mathcal{Q}'} \beta(\tilde{s}),\ \mathcal{L}(\Gamma)=\mathcal{L}(\Delta')\}} \mathcal{Q}';$$

3. $\forall s', \tilde{s}' \in S_{G'}$ if $s' \xrightarrow{\Gamma'}\!\!\!\!\rightarrow_{\mathcal{P}'} \tilde{s}'$ then $\exists \Gamma, \mathcal{P}$ such that $\beta^{-1}(s') \xrightarrow{\Gamma}\!\!\!\!\rightarrow_{\mathcal{P}} \beta^{-1}(\tilde{s}'),\ \mathcal{L}(\Gamma) = \mathcal{L}(\Gamma')$ and

$$\sum_{\{\Delta' \mid s' \xrightarrow{\Delta'}\!\!\!\!\rightarrow_{\mathcal{Q}'} \tilde{s}',\ \mathcal{L}(\Gamma)=\mathcal{L}(\Delta')\}} \mathcal{Q}' = \sum_{\{\Delta \mid \beta^{-1}(s') \xrightarrow{\Delta}\!\!\!\!\rightarrow_{\mathcal{Q}} \beta^{-1}(\tilde{s}')),\ \mathcal{L}(\Gamma)=\mathcal{L}(\Delta)\}} \mathcal{Q}.$$

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

5.6. Interrelations of the probabilistic equivalences

Now we 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:

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

Proof:
We present here a sketch of the proof from [49]. It is enough to prove for $\star = s$, since $\star = i$ is a particular case with the interleaving transition relation. Let $\mathcal{R} : G \underline{\leftrightarrow}_{sp} G'$ and $(s_1, s_2) \in \mathcal{R}$. We have $\forall A \in I\!N_f^{\mathcal{L}}\ \forall \widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}\ s_1 \xrightarrow{A}\!\!\!\!\rightarrow_{\mathcal{Q}} \widetilde{\mathcal{H}} \ \Leftrightarrow\ s_2 \xrightarrow{A}\!\!\!\!\rightarrow_{\mathcal{Q}} \widetilde{\mathcal{H}}$. Let $\mathcal{H} = [s_1]_{\mathcal{R}} = [s_2]_{\mathcal{R}}$. We can rewrite this identity as $\mathcal{H} \xrightarrow{A}\!\!\!\!\rightarrow_{\mathcal{Q}} \widetilde{\mathcal{H}}$, since for all states from $\mathcal{H}$ their probabilities of moving into $\widetilde{\mathcal{H}}$ as a result of execution of A coincide. Let $(A_1 \cdots A_n, \mathcal{P}) \in StepProbTraces(G)$. Since $\mathcal{R} : G \underline{\leftrightarrow}_{sp} G'$, we have $s_G \xrightarrow{A_1}\!\!\!\!\rightarrow_{\mathcal{Q}_1} \mathcal{H}_1 \cap DR(G) \xrightarrow{A_2}\!\!\!\!\rightarrow_{\mathcal{Q}_2} \cdots \xrightarrow{A_n}\!\!\!\!\rightarrow_{\mathcal{Q}_n} \mathcal{H}_n \cap DR(G) \ \Leftrightarrow\ s_{G'} \xrightarrow{A_1}\!\!\!\!\rightarrow_{\mathcal{Q}_1} \mathcal{H}_1 \cap DR(G') \xrightarrow{A_2}\!\!\!\!\rightarrow_{\mathcal{Q}_2} \cdots \xrightarrow{A_n}\!\!\!\!\rightarrow_{\mathcal{Q}_n} \mathcal{H}_n \cap DR(G')$. Next, we 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 $\mathcal{Q}_1, \ldots, \mathcal{Q}_n$, i.e., $\prod_{i=1}^n \mathcal{Q}_i = \sum_{\{\Gamma_1, \ldots, \Gamma_n \mid s_G \xrightarrow{\Gamma_1}\!\!\!\!\rightarrow_{\mathcal{P}_1} \cdots \xrightarrow{\Gamma_n}\!\!\!\!\rightarrow_{\mathcal{P}_n} s_n,\ \mathcal{L}(\Gamma_i)=A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^n \mathcal{P}_i$. This result can also be applied to G'.

It is enough to see now that the summation over *all equivalence classes* is the same as that over *all states*, hence, over *all multisets of activities*, since their executions result the states: $\sum_{\{\Gamma_1, \ldots, \Gamma_n \mid s_G \xrightarrow{\Gamma_1}\!\!\!\!\rightarrow_{\mathcal{P}_1} \cdots \xrightarrow{\Gamma_n}\!\!\!\!\rightarrow_{\mathcal{P}_n} s_n,\ \mathcal{L}(\Gamma_i)=A_i\}} \prod_{i=1}^n \mathcal{P}_i = \sum_{\{\mathcal{H}_1, \ldots, \mathcal{H}_n \mid s_G \xrightarrow{A_1}\!\!\!\!\rightarrow_{\mathcal{Q}_1} \cdots \xrightarrow{A_n}\!\!\!\!\rightarrow_{\mathcal{Q}_n} \mathcal{H}_n \cap DR(G)\}} \prod_{i=1}^n \mathcal{Q}_i = \sum_{\{\mathcal{H}_1, \ldots, \mathcal{H}_n \mid s_{G'} \xrightarrow{A_1}\!\!\!\!\rightarrow_{\mathcal{Q}_1} \cdots \xrightarrow{A_n}\!\!\!\!\rightarrow_{\mathcal{Q}_n} \mathcal{H}_n \cap DR(G')\}} \prod_{i=1}^n \mathcal{Q}_i = \sum_{\{\Gamma'_1, \ldots, \Gamma'_n \mid s_{G'} \xrightarrow{\Gamma'_1}\!\!\!\!\rightarrow_{\mathcal{P}'_1} \cdots \xrightarrow{\Gamma'_n}\!\!\!\!\rightarrow_{\mathcal{P}'_n} s'_n,\ \mathcal{L}(\Gamma'_i)=A_i\}} \prod_{i=1}^n \mathcal{P}'_i$.
Thus, $(A_1 \cdots A_n, \mathcal{P}) \in StepProbTraces(G')$ and $StepProbTraces(G) \subseteq StepProbTraces(G')$. The reverse inclusion is proved by symmetry. $\square$

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

Figure 8. Interrelations of the probabilistic equivalences

Proposition 5.3. For dynamic expressions G and G' the following holds:

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

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

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

Theorem 5.2. Let $\leftrightarrow, \twoheadleftarrow\!\!\twoheadrightarrow \in \{\equiv, \underline{\leftrightarrow}, =, \simeq\}$ and $\star, \star\star \in \{\_, ip, sp, sto, ts\}$, where the symbol '_' denotes no subscription of an expression. For dynamic expressions G and G'

$$G \leftrightarrow_\star G' \;\Rightarrow\; G \twoheadleftrightarrow_{\star\star} G'$$

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

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 $\underline{\leftrightarrow}_{\star p}\to\equiv_{\star p},\ \star\in\{i,s\}$, 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 $\simeq\to=_{ts}$ is valid, since the transition system of a dynamic formula is defined based on its isomorphism class.

($\Rightarrow$) An absence of additional nontrivial arrows (not resulting from the combination of the existing ones) 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 $E = (\{a\},\frac{1}{2})\|(\{b\},\frac{1}{2})$ and $E' = ((\{a\},\frac{1}{2});(\{b\},\frac{1}{2}))[]((\{b\},\frac{1}{2});(\{a\},\frac{1}{2}))$. Then $\overline{E}\underline{\leftrightarrow}_{ip}\overline{E'}$, but $\overline{E}\not\equiv_{sp}\overline{E'}$, since only in $TS^*(\overline{E'})$ multiactions $\{a\}$ and $\{b\}$ cannot be executed concurrently.
- Let $E = (\{a\},\frac{1}{2});((\{b\},\frac{1}{2})[](\{c\},\frac{1}{2}))$ and $E' = ((\{a\},\frac{1}{2});(\{b\},\frac{1}{2}))[]((\{a\},\frac{1}{2});(\{c\},\frac{1}{2}))$. Then $\overline{E}\equiv_{sp}\overline{E'}$, but $\overline{E}\underline{\not\leftrightarrow}_{ip}\overline{E'}$, since only in $TS^*(\overline{E'})$ a multiaction $\{a\}$ can be executed so that no multiaction $\{b\}$ can occur afterwards.
- Let $E = (\{a\},\frac{1}{2});(\{b\},\frac{1}{2})$ and $E' = (\{a\},\frac{1}{2});(\{b\},\frac{1}{2})[](\{a\},\frac{1}{2});(\{b\},\frac{1}{2})$. Then $\overline{E}\underline{\leftrightarrow}_{sp}\overline{E'}$, but $\overline{E}\neq_{sto}\overline{E'}$, since only in $TS^*(\overline{E'})$ 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 $TS^*(\overline{E'})$. Note that in $TS^*(\overline{E'})$, the only transition with the same multiaction part of label has probability $\frac{1}{2}$.
- Let $E = (\{a\},\frac{1}{2})$ and $E' = (\{a\},\frac{1}{2})[](\{a\},\frac{1}{2})$. Then $\overline{E} =_{sto}\overline{E'}$, but $\overline{E}\neq_{ts}\overline{E'}$, since only $TS(\overline{E'})$ has two transitions.
- Let $E = (\{a\},\frac{1}{2})$ and $E' = ((\{a\},\frac{1}{2});(\{\hat{a}\},\frac{1}{2}))$ sy a. Then $\overline{E} =_{ts}\overline{E'}$, but $\overline{E}\not\simeq\overline{E'}$, since $\overline{E}$ and $\overline{E'}$ cannot be reached from each other by applying inaction rules.

$\square$

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., $N = Box_{dts}(\overline{E})$ and $N' = Box_{dts}(\overline{E'})$ for each picture (a)–(e). Since all the equivalences of dynamic expressions can be transferred to the corresponding marked dts-boxes, we depict also the net analogues (denoted by the same symbols) of the algebraic equivalences which relate the nets.

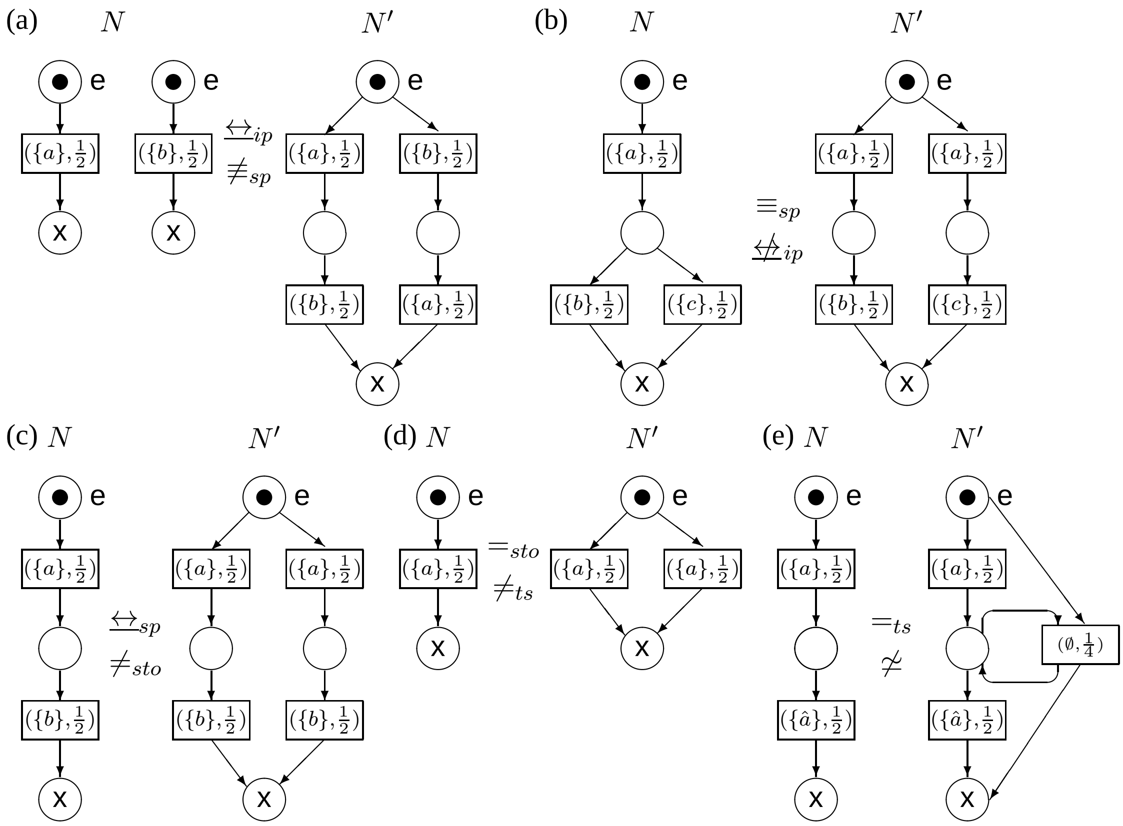

Figure 9. Dts-boxes of the dynamic expressions from equivalence examples of Theorem 5.2

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. A consistency 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 in the definitions of the equivalences, i.e., from the activities with empty multiaction part. The abstraction from empty loops and that from silent activities could be done in one step as well. The main point here is that we should collect probabilities during the abstraction from an internal activity. As a result, we shall have the algebraic analogues of the net probabilistic equivalences from [13, 14]. Moreover, we plan to extend $dtsPBC$ with infiniteness constructs such as iteration and recursion. The difficulty here is a proper handle the infinite summation and multiplication of step probabilities as well as a safety of the dts-boxes resulting from expressions specifying loops.

Acknowledgements I thank E. Best for supervision and encouraging discussions during my work at the University of Oldenburg, Germany. I am also grateful to V.R. Valero and H.S. Macià for many valuable advices as well as to their colleagues from the High Polytechnic School of Albacete, Spain, for presentations and kind hospitality. Many thanks to the anonymous referees for their helpful comments.

References

[1] Andova, S.: Process algebra with probabilistic choice, Lect. Notes Comp. Sci., vol. 1601 of *Lect. Notes Comp. Sci.*, Springer, 1999, 111–129.

[2] Bernardo, M., Gorrieri, R.: A tutorial on EMPA: a theory of concurrent processes with nondeterminism, priorities, probabilities and time, *Theor. Comput. Sci.*, **202**, 1998, 1–54, http://www.sti.uniurb.it/bernardo/documents/tcs202.pdf.

[3] Best, E., Devillers, R., Esparza, J.: General refinement and recursion operations in the box calculus, Lect. Notes Comp. Sci., vol. 665 of *Lect. Notes Comp. Sci.*, Springer, 1993, 130–140.

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

[5] Best, E., Devillers, R., Koutny, M.: Petri nets, process algebras and concurrent programming languages, Lect. Notes Comp. Sci., vol. 1492 of *Lect. Notes Comp. Sci.*, Springer, 1998, 1–84, http://parsys.informatik.uni-oldenburg.de/~best/publications/apnf.ps.gz.

[6] Best, E., Devillers, R., Koutny, M.: *Petri net algebra*, EATCS Monographs on Theor. Comput. Sci., Springer, 2001, 378 pages.

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

[8] Best, E., Koutny, M.: A refined view of the box algebra, *Proc.* 16^{th} *ICATPN 1995*, Lect. Notes Comp. Sci., vol. 935 of *Lect. Notes Comp. Sci.*, Springer, 1995, 1–20, http://parsys.informatik.uni-oldenburg.de/~best/publications/pn95.ps.gz.

[9] Buchholz, P.: Markovian process algebra: composition and equivalence, *Proc.* 2^{nd} *Int. Workshop on Process Algebras and Performance Modelling (PAPM) 1994*, Arbeitsberichte des IMMD, number 27 in Arbeitsberichte des IMMD, University of Erlangen, Germany, 1994, 11–30.

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

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

[12] Buchholz, P., Kemper, P.: Quantifying the dynamic behavior of process algebras, Lect. Notes Comp. Sci., vol. 2165 of *Lect. Notes Comp. Sci.*, Springer, 2001, 184–199.

[13] Buchholz, P., Tarasyuk, I. V.: *A class of stochastic Petri nets with step semantics and related equivalence notions*, Technische Berichte TUD-FI00-12, Fakultät Informatik, Technische Universität Dresden, Germany, 2000, 18 pages, ftp://ftp.inf.tu-dresden.de/pub/berichte/tud00-12.ps.gz.

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

[15] Devillers, R.: Petri boxes and finite processes, Lect. Notes Comp. Sci., vol. 1119 of *Lect. Notes Comp. Sci.*, Springer, 1996, 465–480.

[16] Donatelli, S., Ribaudo, M., Hillston, J.: A comparison of perfomance evaluation process algebra and generalized stochastic Petri nets, *Proc.* 6^{th} *Int. Workshop on Petri Nets and Performance Models (PNPM) 1995*, IEEE Computer Society Press, Durham, USA, 1995, 158–168.

[17] Florin, G., Natkin, S.: Les reseaux de Petri stochastiques, *Technique et Science Informatique*, **4**, 1985, 143–160.

[18] van Glabbeek, R. J.: The linear time – branching time spectrum II: the semantics of sequential systems with silent moves. Extended abstract, *Proc.* 4^{th} *CONCUR 1993*, Lect. Notes Comp. Sci., vol. 715 of *Lect. Notes Comp. Sci.*, Springer, 1993, 66–81.

[19] Hansson, H.: Time and probability in formal design of distributed systems, *Real-Time Safety Critical Systems*, **1**, 1994.

[20] Hermanns, H., Rettelbach, M.: Syntax, semantics, equivalences and axioms for MTIPP, *Proc.* 2^{nd} *Int. Workshop on Process Algebras and Performance Modelling (PAPM) 1994*, Arbeitsberichte des IMMD, number 27 in Arbeitsberichte des IMMD, University of Erlangen, Germany, 1994, 71–88, http://ftp.informatik.uni-erlangen.de/local/inf7/papers/Hermanns/syntax_semantics_ equivalences_axioms_for_MTIPP.ps.gz.

[21] Hillston, J.: *A compositional approach to performance modelling*, Cambridge University Press, UK, 1996, 158 pages, http://www.dcs.ed.ac.uk/pepa/book.pdf.

[22] Jonsson, B., Yi, W., Larsen, K. G.: Probabilistic extensions of process algebras, in: *Handbook of Process Algebra* (J. A. Bergstra, A. Ponse, S. A. Smolka, Eds.), chapter 11, Elsevier Science B.V., Amsterdam, The Netherlands, 2001, 685–710.

[23] Kotov, V. E.: An algebra for parallelism based on Petri nets, Lect. Notes Comp. Sci., vol. 64 of *Lect. Notes Comp. Sci.*, Springer, 1978, 39–55.

[24] Kotov, V. E., A.Cherkasova, L.: On structural properties of generalized processes, Lect. Notes Comp. Sci., vol. 188 of *Lect. Notes Comp. Sci.*, Springer, 1985, 288–306.

[25] Koutny, M.: Partial order semantics of box expressions, Lect. Notes Comp. Sci., vol. 815 of *Lect. Notes Comp. Sci.*, Springer, 1994, 318–337.

[26] Koutny, M., Best, E.: Operational and denotational semantics for the box algebra, *Theor. Comput. Sci.*, **211**, 1999, 1–83, http://parsys.informatik.uni-oldenburg.de/~best/publications/tcs.ps.gz.

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

[28] Larsen, K. G., Skou, A.: Compositional verification of probabilistic processes, Lect. Notes Comp. Sci., vol. 630 of *Lect. Notes Comp. Sci.*, Springer, 1992, 456–471.

[29] Macià, H.: *sPBC: Una extensión Markoviana del Petri box calculus*, Technical report, Departamento de Informática, Universidad de Castilla-La Mancha, Albacete, Spain, 2003, Ph. D. thesis, 249 pages, in Spanish, http://www.info-ab.uclm.es/retics/publications/2003/sPBCthesis03.pdf.

[30] Macià, H., Valero, V., Cazorla, D., Cuartero, F.: Introducing the iteration in sPBC, *Proc.* 24^{th} *FORTE 2004*, Lect. Notes Comp. Sci., vol. 3235 of *Lect. Notes Comp. Sci.*, Springer, 2004, 292–308, http://www.info-ab.uclm.es/retics/publications/2004/forte04.pdf.

[31] Macià, H., Valero, V., Cuartero, F.: *A congruence relation in finite sPBC*, Technical Report DIAB-02-01-31, Departamento de Informática, Universidad de Castilla-La Mancha, Albacete, Spain, 2002, 34 pages, http://www.info-ab.uclm.es/retics/publications/2002/tr020131.ps.

[32] Macià, H., Valero, V., Cuartero, F.: Defining equivalence relations in sPBC, *Proc.* 1^{st} *Int. Conf. on the Principles of Software Engineering (PriSE) 2004*, Buenos Aires, Argentina, 2004, 195–205, http://www.info-ab.uclm.es/retics/publications/2004/prise04.pdf.

[33] Macià, H., Valero, V., Cuartero, F., de Frutos, D.: A congruence relation for sPBC, 2004, 62 pages, submitted.

[34] Macià, H., Valero, V., Cuartero, F., Pelayo, F.: Improving the synchronization in stochastic Petri box calculus, *Actas de las II Jornadas sobre Programacion y Lenguajes (PROLE) 2002*, El Escorial, Spain, 2002.

[35] Macià, H., Valero, V., Cuartero, F., Pelayo, F.: *A new proposal for the synchronization in sPBC*, Technical Report DIAB-02-01-26, Departamento de Informática, Universidad de Castilla-La Mancha, Albacete, Spain, 2002, 15 pages, http://www.info-ab.uclm.es/sec-ab/Tecrep/newproposalsysPBC.ps.

[36] Macià, H., Valero, V., Cuartero, F., Pelayo, F.: A new synchronization in finite stochastic Petri box calculus, *Proc.* 3^{rd} *International IEEE Conference on Application of Concurrency to System Design*, IEEE Computer Society Press, Guimarães, Portugal, 2003, 216–225, http://www.info-ab.uclm.es/retics/publications/2003/acsd03.pdf.

[37] Macià, H., Valero, V., Cuartero, F., Ruiz, M. C.: sPBC: a Markovian extension of Petri box calculus with immediate multiactions, 2005, 27 pages, work in progress.

[38] Macià, H., Valero, V., D., C., Cuartero, F.: *Introducing the iteration in sPBC*, Technical Report DIAB-03-01-37, Departamento de Informática, Universidad de Castilla-La Mancha, Albacete, Spain, 2003, 20 pages, http://www.info-ab.uclm.es/descargas/tecnicalreports/DIAB-03-01-37/diab030137.zip.

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

[40] Macià, H., Valero, V., de Frutos, D.: sPBC: a Markovian extension of finite PBC, *Actas de IX Jornadas de Concurrencia (JC) 2001*, Sitges, Spain, 2001, 243–256, http://www.info-ab.uclm.es/retics/publications/2001/mvfjc01.ps.

[41] Macià, H., Valero, V., de Frutos, D., Cuartero, F.: Extending PBC with Markovian multiactions, *Proc. XXVII Conferencia Latinoamericana de Informática (CLEI) 2001* (J. A. Montilva, I. Besembel, Eds.), Universidad de los Andes, Mérida, Venezuela, 2001, 12 pages, http://www.info-ab.uclm.es/retics/publications/2001/clei01.ps.

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

[43] Molloy, M. K.: Performance analysis using stochastic Petri nets, *IEEE Transactions on Computing*, **31**, 1982, 913–917.

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

[45] Núñez, M.: An axiomatization of probabilistic testing, Lect. Notes Comp. Sci., vol. 1601 of *Lect. Notes Comp. Sci.*, Springer, 1999, 130–150, http://dalila.sip.ucm.es/miembros/manolo/papers/arts99.ps.gz.

[46] Núñez, M., de Frutos, D., Llana, L.: Acceptance trees for probabilistic processes, Lect. Notes Comp. Sci., vol. 962 of *Lect. Notes Comp. Sci.*, Springer, 1995, 249–263, http://dalila.sip.uclm.es/membros/manolo/papers/concur95.ps.gz.

[47] Ribaudo, M.: Stochastic Petri net semantics for stochastic process algebra, *Proc.* 6^{th} *Int. Workshop on Petri Nets and Performance Models (PNPM) 2001*, IEEE Computer Society Press, Durham, USA, 1995, 148–157.

[48] Tarasyuk, I. V.: Equivalence notions applied to designing concurrent systems with the use of Petri nets, *Programming and Computer Software*, **24**, 1998, 162–175.

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