Bull. Nov. Comp. Center, Comp. Science, 28 (2008), 121–141

A notion of congruence for *dtsPBC**

I. V. Tarasyuk

Abstract. Algebra *dtsPBC* is a discrete time stochastic extension of finite Petri box calculus (*PBC*) enriched with iteration. In this paper, we define a number of stochastic equivalences for *dtsPBC* which allow one to identify finite and infinite stochastic processes with similar behaviour. A problem of preservation of the equivalences by algebraic operations is discussed. As a result, we construct an equivalence that is a congruence relation.

Keywords: stochastic process algebras, Petri box calculus, discrete time, operational semantics, denotational semantics, empty loops, stochastic equivalences, congruence.

1. Introduction

Algebraic process calculi is a well-known formal model for specification of computing systems and analysis of their behaviour. In such process algebras (PAs), systems and processes are specified by formulas, and verification of their properties is accomplished at a syntactic level by means of equivalences, axioms and inference rules. In the last decades, stochastic extensions of PAs were proposed and became widespread. Stochastic process algebras (SPAs) do not just specify actions which can happen as usual process algebras (qualitative features), but they associate some quantitative parameters with actions (quantitative characteristics). The most popular SPAs proposed so far are *TIPP* [4], *PEPA* [3] and *EMPA* [2].

Petri box calculus (*PBC*) [1] is a flexible and expressive process algebra based on calculus *CCS* [5]. It was developed as a tool for specification of Petri nets structure and their interrelations. Its goal was also to propose a compositional semantics for high level constructs of concurrent programming languages in terms of elementary Petri nets. *PBC* has a step operational semantics in terms of labeled transition systems based on the Structured Operational Semantics (SOS) rules. Its denotational semantics was proposed in terms of a subclass of Petri nets (PNs) equipped with interface and considered up to isomorphism called Petri boxes.

A stochastic extension of *PBC* called stochastic Petri box calculus (*sPBC*) was proposed in [9, 10]. Only a finite part of *PBC* was used for the

* This work was supported in part by Deutscher Akademischer Austauschdienst (DAAD), grant A/08/08590, Deutsche Forschungsgemeinschaft (DFG), grant 436 RUS 113/1002/01, and Russian Foundation for Basic Research (RFBR), grant 09-01-91334

stochastic enrichment, i.e., $sPBC$ has neither refinement nor recursion nor iteration operations. The calculus has an interleaving operational semantics in terms of labeled transition systems. Its denotational semantics was defined in terms of a subclass of labeled continuous time stochastic PNs (LCTSPNs) called stochastic Petri boxes (s-boxes). The results on constructing the iteration for $sPBC$ were reported in [6, 7]. In [8], a congruence relation for $sPBC$ was constructed.

In [11, 13], a discrete time stochastic extension $dtsPBC$ of finite PBC was presented. A step operational semantics of $dtsPBC$ was constructed with the use of labeled probabilistic transition systems. Its denotational semantics was defined based on a subclass of labeled discrete time stochastic PNs (LDTSPNs) called discrete time stochastic Petri boxes (dts-boxes). A variety of probabilistic equivalences were proposed to identify stochastic processes with similar behaviour which are differentiated by the semantic equivalence. The interrelations of all the introduced equivalences were studied. In [12], the iteration operator was added to the syntax of $dtsPBC$ to specify infinite processes.

In this paper, a problem of preservation of the equivalence notions by algebraic operations is discussed. First, we present the syntax of the extended $dtsPBC$. Second, we describe the operational and denotational semantics of the calculus. Further, we consider a number of stochastic algebraic equivalences based on transition systems without empty behaviour and present a diagram of equivalence interrelations. The proposed equivalences are then used to construct a new congruence relation for the algebra.

The paper is organized as follows. In the next Section 2, the syntax of the algebra $dtsPBC$ is presented. In Section 3, we describe its operational semantics in terms of labeled transition systems. In Section 4, we present a short overview of the denotational semantics of the algebra based on a subclass of LDTSPNs. Section 5 is devoted to the construction and the interrelations of stochastic algebraic equivalences. Preservation of the equivalences by the algebraic operations, i.e., a congruence problem, is discussed in Section 6. The concluding Section 7 summarizes the results obtained and outlines research perspectives in this area.

2. Syntax

In this section, we propose the syntax of the discrete time stochastic extension of finite PBC enriched with iteration called *discrete time stochastic Petri box calculus* ($dtsPBC$).

We denote the *set of all finite multisets* over X by $I\!N_f^X$. Let $Act = \{a, b, \ldots\}$ be the set of *elementary actions*. Then $\widehat{Act} = \{\hat{a}, \hat{b}, \ldots\}$ is the set of *conjugated actions (conjugates)* such that $a \neq \hat{a}$ 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 the probabilities of state changes (steps) at discrete time moments. 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, *synchronization* sy and *iteration* $[**]$.

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 correspond to unmarked SPNs.

Definition 1. Let $(\alpha, \rho) \in \mathcal{SL}$ and $a \in Act$. A *static expression* of *dtsPBC* is defined as

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

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

To avoid inconsistency of the iteration operator, we should not allow any concurrency in the highest level of the second argument of iteration. This is not a severe restriction though, since we can always prefix parallel expressions by an activity with the empty multiaction and appropriate probability.

Definition 2. Let $(\alpha, \rho) \in \mathcal{SL}$ and $a \in Act$. A *regular static expression* of *dtsPBC* is defined as

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

RegStatExpr denotes the set of *all regular static expressions* of *dtsPBC*.

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

Definition 3. Let $(\alpha, \rho) \in \mathcal{SL},\ a \in Act$ and $E \in RegStatExpr$. A *regular dynamic expression* of *dtsPBC* is defined as

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

RegDynExpr denotes the set of *all regular dynamic expressions* of *dtsPBC*. We shall consider regular expressions only, hence, we can omit the word "regular".

3. Operational semantics

In this section, we present a short overview of the step operational semantics. It was defined in [12] via labeled transition systems based on transformation rules for dynamic expressions.

Note that expressions can contain identical activities. To avoid technical difficulties, we can always enumerate coinciding activities from left to right in the syntax of expressions.

The *inaction rules* describe expression transformations due to execution of the empty multiset of activities (semantics-preserving syntactic transformations) and have a form $G \stackrel{\emptyset}{\rightarrow} \widetilde{G}$, where $G, \widetilde{G} \in RegDynExpr$. The only non-standard inaction rule (comparing with *PBC*) is $G \stackrel{\emptyset}{\rightarrow} G$.

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

Definition 4. Let $\simeq \, = (\stackrel{\emptyset}{\rightarrow} \cup \stackrel{\emptyset}{\leftarrow})^*$ be isomorphism of dynamic expressions in *dtsPBC*. 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.

The *action rules* describe expression transformations due to the execution of non-empty multisets of activities. The rules have a form $G \stackrel{\Gamma}{\rightarrow} \widetilde{G}$, where $G \in OpRegDynExpr$, $\widetilde{G} \in RegDynExpr$ and $\Gamma \in I\!N_f^{\mathcal{SL}} \setminus \emptyset$. The only non-standard action rule is $\frac{G \text{ sy } a \stackrel{\Gamma + \{(\alpha,\rho)\} + \{(\beta,\chi)\}}{\longrightarrow} \widetilde{G} \text{ sy } a,\ a \in \mathcal{A}(\alpha),\ \hat{a} \in \mathcal{A}(\beta)}{G \text{ sy } a \stackrel{\Gamma + \{(\alpha \oplus_a \beta, \rho \cdot \chi)\}}{\longrightarrow} \widetilde{G} \text{ sy } a}$.

Definition 5. 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 isomorphism class). The *derivation set* of a dynamic expression G, denoted by $DR(G)$, is the minimal set such that

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

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

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

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

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

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

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

The probability that the activities from Γ *happen* in s is

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

The probability that the execution of *any* activities changes s to $\tilde{s}$ is

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

Definition 6. Let G be a dynamic expression. The *(labeled probabilistic) transition system* of G is a quadruple $TS(G) = (S_G, L_G, \mathcal{T}_G, s_G)$, where

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

A transition $(s, (\Gamma, \mathcal{P}), \tilde{s}) \in \mathcal{T}_G$ will be written as $s \xrightarrow{\Gamma}_{\mathcal{P}} \tilde{s}$. It is interpreted as follows: the probability to change the state s to $\tilde{s}$ in the 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 $\exists \mathcal{P}\ s \xrightarrow{\Gamma}_{\mathcal{P}} \tilde{s}$ and $s \rightarrow \tilde{s}$ if $\exists \Gamma\ s \xrightarrow{\Gamma} \tilde{s}$.

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 isomorphism classes. We have to keep track of such executions called *empty loops*, because they have nonzero probabilities. This follows from the definition of $PF(\emptyset, s)$ and the fact that multiaction probabilities cannot be equal to 1 as they belong to the interval $(0; 1)$.

Definition 7. Let G, G' be dynamic expressions and $TS(G) = (S_G, L_G, \mathcal{T}_G, s_G)$, $TS(G') = (S_{G'}, L_{G'}, \mathcal{T}_{G'}, s_{G'})$ be their transition systems. A mapping $\beta : S_G \rightarrow S_{G'}$ is an *isomorphism* between $TS(G)$ and $TS(G')$, denoted by $\beta : TS(G) \simeq TS(G')$, if

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

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

For $E \in$ *RegStatExpr*, let $TS(E) = TS(\overline{E})$.

Definition 8. 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 9. 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 the transitions $s \rightarrow_{\mathcal{P}} \tilde{s}$, if $s \rightarrow \tilde{s}$ and $\mathcal{P} = PM(s, \tilde{s})$.

For $E \in$ *RegStatExpr*, let $DTMC(E) = DTMC(\overline{E})$.

4. Denotational semantics

In this section, we present a short overview of the denotational semantics. It was defined in [12] via a subclass of labeled DTSPNs called discrete time stochastic Petri boxes (dts-boxes).

Definition 10. 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).

Let $t \in T_N,\ U \in I\!N_f^{T_N}$. 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 $U^\bullet = \sum_{t\in U} t^\bullet$. We require that $\forall t \in T_N\ {}^\bullet t \neq \emptyset \neq t^\bullet$ and ${}^\bullet t \cap t^\bullet = \emptyset$.

In addition, for the set of *entry* places of N defined as ${}^\circ N = \{p \in P_N \mid \Lambda_N(p) = \mathsf{e}\}$ and the set of *exit* places of N defined as $N^\circ = \{p \in P_N \mid \Lambda_N(p) = \mathsf{x}\}$ we require that ${}^\circ N \neq \emptyset \neq N^\circ$ and ${}^\bullet({}^\circ N) = \emptyset = (N^\circ)^\bullet$.

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 denote $\overline{N} = (N, {}^\circ N)$ and $\underline{N} = (N, N^\circ)$. 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 $\forall t \in T_N\ \Omega_N(t) = \Omega(\Lambda_N(t))$ (*transition probability* function) and $L_N(t) = \mathcal{L}(\Lambda_N(t))$ (*transition labeling* function).

Let $N_{(\alpha,\rho)_i}$ denote the plain dts-box of the enumerated elementary static expression $(\alpha,\rho)_i$ and $\Theta_\circ(N_1, N_2),\ \circ \in \{;, [], \|\},\ \Theta_{[f]}(N_1),\ \Theta_{\circ a}(N_1),\ \circ \in \{\mathsf{rs}, \mathsf{sy}\},\ \Theta_{[**]}(N_1)$ denote the composite plain dts-boxes resulted from application of the net algebraic operations to the plain dts-boxes N_1 and N_2.

Definition 11. Let $(\alpha,\rho) \in \mathcal{SL},\ a \in Act$ and $E, F, K \in RegStatExpr$. The *denotational semantics* of *dtsPBC* is a mapping Box_{dts} from *RegStatExpr* into the area of plain dts-boxes defined as follows:

1. $Box_{dts}((\alpha,\rho)_i) = N_{(\alpha,\rho)_i}$;
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. $Box_{dts}(E \circ a) = \Theta_{\circ a}(Box_{dts}(E)),\ \circ \in \{\mathsf{rs}, \mathsf{sy}\}$;
5. $Box_{dts}([E * F * K]) = \Theta_{[**]}(Box_{dts}(E), Box_{dts}(F), Box_{dts}(K))$.

For $E \in RegStatExpr$, 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. Note that in this case, the names of transitions of the dts-box corresponding to a static expression could be identified with the enumerated activities of the latter.

For a dts-box N, we denote its *reachability graph* by $RG(N)$ and its *underlying DTMC* by $DTMC(N)$.

Theorem 1. *[12] For any static expression E*

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

Proposition 1. *[12] For any static expression E*

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

5. Stochastic equivalences

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

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

5.1. Empty loops

Let G be a dynamic expression. A transition system $TS(G)$ can have loops going from a state to itself which are labeled by the empty set and have non-zero probability. Such *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 $s \in DR(G)$.

The *probability to stay in* s *due to* k $(k \geq 1)$ *empty loops* is $(PT(\emptyset, s))^k$.

The *probability to execute in* s *a non-empty multiset of activities* $\Gamma \in Exec(s) \setminus \{\emptyset\}$ *after possible empty loops* is

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

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

Definition 12. The *(labeled probabilistic) transition system without empty loops* $TS^*(G)$ has the state space $DR(G)$ and the transitions $s \overset{\Gamma}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s}$, if $s \overset{\Gamma}{\rightarrow} \tilde{s}$, $\Gamma \neq \emptyset$ and $\mathcal{P} = PT^*(\Gamma, s)$.

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

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

For $E \in RegStatExpr$, let $TS^*(E) = TS^*(\overline{E})$.

Definition 13. 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 14. The *underlying DTMC without empty loops* $DTMC^*(G)$ has the state space $DR(G)$ and the transitions $s \twoheadrightarrow_{\mathcal{P}} \tilde{s}$, if $s \twoheadrightarrow \tilde{s}$, where $\mathcal{P} = PM^*(s, \tilde{s})$ and

$$PM^*(s, \tilde{s}) = \sum_{\{\Gamma \mid s \overset{\Gamma}{\twoheadrightarrow} \tilde{s}\}} PT^*(\Gamma, s).$$

For $E \in RegStatExpr$, let $DTMC^*(E) = DTMC^*(\overline{E})$.

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

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

Let $N = (P_N, T_N, W_N, \Omega_N, L_N, M_N)$ ba a LDTSPN and $M, \widetilde{M} \in I\!N_f^{P_N}$, $t \in T_N$, $U \subseteq T_N$. Then the transition relations $M \overset{U}{\twoheadrightarrow}_{\mathcal{P}} \widetilde{M}$, $M \overset{U}{\twoheadrightarrow} \widetilde{M}$, $M \rightarrow\rightarrow \widetilde{M}$, $M \overset{t}{\twoheadrightarrow}_{\mathcal{P}} \widetilde{M}$, $M \overset{t}{\twoheadrightarrow} \widetilde{M}$, $M \twoheadrightarrow_{\mathcal{P}} \widetilde{M}$, $M \overset{t}{\rightarrowtail}_{\mathcal{P}} \widetilde{M}$, the *reachability graph without empty loops* $RG^*(N)$ and the *underlying DTMC without empty loops* $DTMC^*(N)$ are defined like the corresponding notions for dynamic expressions.

Theorem 2. *For any static expression E*

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

Proof. As for the qualitative (functional) behaviour, we have the same isomorphism as in PBC. The quantitative behaviour is the same by the following reasons. First, the activities of an expression have probability parts coinciding with the probabilities of the transitions belonging to the corresponding dts-box. Second, both in stochastic processes specified by expressions and in dts-boxes conflicts are resolved via the same probability functions used to construct the corresponding transition systems and reachability graphs.

Proposition 2. *For any static expression E*

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

Proof. By Theorem 2 and definitions of underlying DTMC for dynamic expressions and LDTSPNs, since transition probabilities of the associated DTMCs are the sums of those belonging to transition systems or reachability graphs.

Theorem 2 guarantees that the net versions of algebraic equivalences could be easily defined. For every equivalence on the transition system without empty loops of a dynamic expression, a similarly defined analogue exists on the reachability graph without empty loops of the corresponding dts-box.

5.2. Stochastic trace equivalences

Trace equivalences are the least discriminating ones. In the trace semantics, the behaviour 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.

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

Definition 15. An *interleaving stochastic trace* of a dynamic expression G is a pair $(\sigma, PT^*(\sigma))$, where $\sigma = \alpha_1 \cdots \alpha_n \in \mathcal{L}^*$ and

$$PT^*(\sigma) = \sum_{\{(\alpha_1,\rho_1),\ldots,(\alpha_n,\rho_n) \mid [G]_\simeq = s_0 \stackrel{(\alpha_1,\rho_1)}{\twoheadrightarrow} \cdots \stackrel{(\alpha_n,\rho_n)}{\twoheadrightarrow} s_n\}} \prod_{i=1}^{n} PT_i^*((\alpha_i,\rho_i), s_{i-1}).$$

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

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

Definition 16. A *step stochastic trace* of a dynamic expression G is a pair $(\Sigma, PT^*(\Sigma))$, where $\Sigma = A_1 \cdots A_n \in (\mathbb{N}_f^{\mathcal{L}})^*$ and

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

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

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

5.3. Stochastic bisimulation equivalences

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

Let G be a dynamic expression and $\mathcal{H} \subseteq DR(G)$. Then for some $s \in DR(G)$ and $A \in \mathbb{N}_f^{\mathcal{L}}$ we write $s \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}} \mathcal{H}$, where $\mathcal{P} = PM_A^*(s, \mathcal{H})$ and

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

Thus, $PM_A^*(s,\mathcal{H})$ is the overall probability to come into the set of states $\mathcal{H}$ starting in s via steps with the multiaction part A. The summation above reflects the probability of the events union.

We propose the *interleaving transition relation* $s \stackrel{\alpha}{\rightharpoondown}_{\mathcal{P}} \mathcal{H}$, where $\mathcal{P} = PM_{i\alpha}^*(s,\mathcal{H})$ and

$$PM_{i\alpha}^*(s,\mathcal{H}) = \sum_{\{(\alpha,\rho) \mid \exists \tilde{s} \in \mathcal{H}\ s \stackrel{(\alpha,\rho)}{\twoheadrightarrow} \tilde{s}\}} PT_i^*((\alpha,\rho), s).$$

Definition 17. Let G be a dynamic expression. An *equivalence* relation $\mathcal{R} \subseteq DR(G)^2$ is a $\star$-*stochastic bisimulation* between states s_1 and s_2 from $DR(G)$, $\star \in$\{*interleaving, step*\}, denoted by $\mathcal{R} : s_1 \underline{\leftrightarrow}_{\star s} s_2,\ \star \in \{i, s\}$, if $\forall \mathcal{H} \in DR(G)/_{\mathcal{R}}$

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

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

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

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

Definition 18. Let G, G' be dynamic expressions. A relation $\mathcal{R} \subseteq (DR(G) \cup DR(G'))^2$ is a $\star$-*stochastic bisimulation* between G and G', $\star \in$\{*interleaving, step*\}, denoted by $\mathcal{R} : G \underline{\leftrightarrow}_{\star s} G'$, if $\mathcal{R} : [G]_{\simeq} \underline{\leftrightarrow}_{\star s} [G']_{\simeq},\ \star \in \{i, s\}$.

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

5.4. 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 collect the probabilities of all transitions between the same pair of states such that the transition labels have the same multiaction parts.

Let G be a dynamic expression and $s, \tilde{s} \in DR(G)$ such that $s \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}} \{\tilde{s}\}$. In this case, we write $s \stackrel{A}{\twoheadrightarrow}_{\mathcal{P}} \tilde{s}$.

Definition 19. Let G, G' be dynamic expressions. A mapping $\beta : DR(G) \rightarrow DR(G')$ is a *stochastic isomorphism* between G and G', denoted by $\beta : G =_{sto} G'$, if

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

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

5.5. Interrelations of the stochastic equivalences

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

Proposition 3. *For dynamic expressions G and G' the following holds:*

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

Proof. Analogous to the proof of the corresponding proposition from [13], but for infinite processes as well.

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

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

Example 1. Let $E = (\{a\}, \frac{1}{2})$ and $E' = (\{a\}, \frac{1}{2})_1[](\{a\}, \frac{1}{2})_2$. Then $\overline{E} =_{sto} \overline{E'}$, but $\overline{E}$ is not equivalent to $\overline{E'}$ according to the stronger version of stochastic isomorphism, since the probability of the only non-empty transition in $TS(\overline{E})$ is $\frac{1}{2}$, whereas the probability of both non-empty transitions in $TS(\overline{E'})$ is $\frac{1}{3}$, and $\frac{1}{2} \neq \frac{1}{3} + \frac{1}{3}$. On the other hand, the probability of the only non-empty transition in $TS^*(\overline{E})$ is 1, the probability of both non-empty

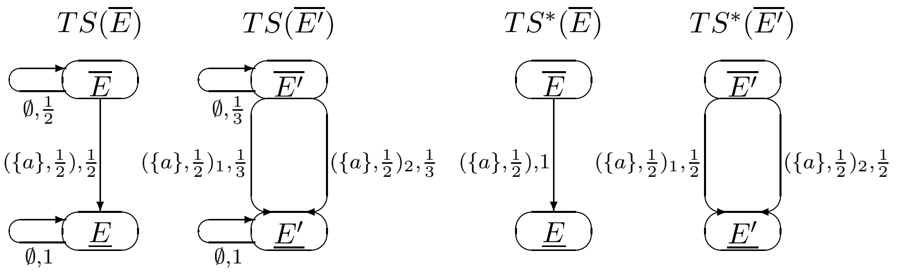

Figure 1. A problem with the stochastic isomorphism based on transition systems with empty loops

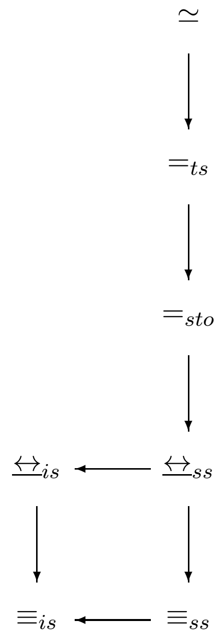

Figure 2. Interrelations of the stochastic equivalences

transitions in $TS^*(\overline{E'})$ is $\frac{1}{2}$, and $1 = \frac{1}{2} + \frac{1}{2}$. The transition systems with and without empty loops of $\overline{E}$ and $\overline{E}'$ are presented in Figure 1.

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

Theorem 3. *Let* $\leftrightarrow, \leftrightarrow\!\!\!\!\leftrightarrow \in \{\equiv, \underline{\leftrightarrow}, =, \simeq\}$ *and* $\star, \star\star \in \{\_, is, ss, sto, ts\}$*. For dynamic expressions* G *and* G'

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

iff there exists a directed path from $\leftrightarrow_\star$ *to* $\leftrightarrow\!\!\!\!\leftrightarrow_{\star\star}$ *in the graph in Figure 2.*

Proof. Analogous to the proof of the corresponding theorem from [13], but for infinite processes as well.

6. Preservation by algebraic operations

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

Definition 20. Let $\leftrightarrow$ be an equivalence of dynamic expressions. Two static expressions E and E' are *equivalent with respect to* $\leftrightarrow$, denoted by $E \leftrightarrow E'$, if $\overline{E} \leftrightarrow \overline{E'}$.

Let us investigate which algebraic equivalences we proposed are congruences on static expressions. The following example demonstrates that no equivalence between $\equiv_{is}$ and $=_{sto}$ is a congruence.

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

In Figure 3 the marked dts-boxes corresponding to the dynamic expressions above are presented, i.e., $N_1 = Box_{dts}(\overline{E}),\ N_1' = Box_{dts}(\overline{E'}),\ N_2 = Box_{dts}(\overline{F})$ and $N = Box_{dts}(\overline{E[]F}),\ N' = Box_{dts}(\overline{E'[]F})$. In addition, we depict the net analogues of the algebraic equivalences.

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

Proposition 4. *Let* $\star \in \{i, s\},\ \star\star \in \{sto, ts\}$. *The equivalences* $\equiv_\star$, $\underline{\leftrightarrow}_\star$, $=_{\star\star}$ *are not preserved by algebraic operations.*

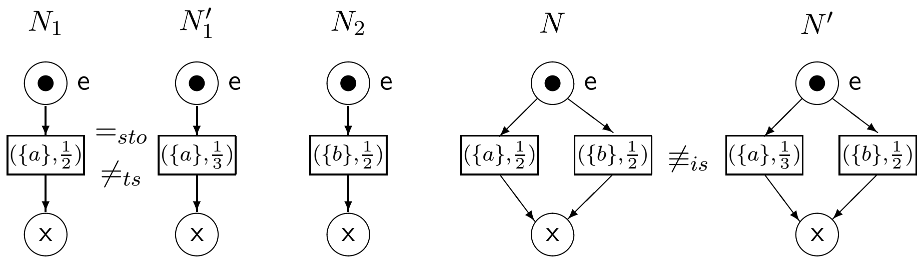

Figure 3. The equivalences between $\equiv_{is}$ and $=_{sto}$ are not congruences

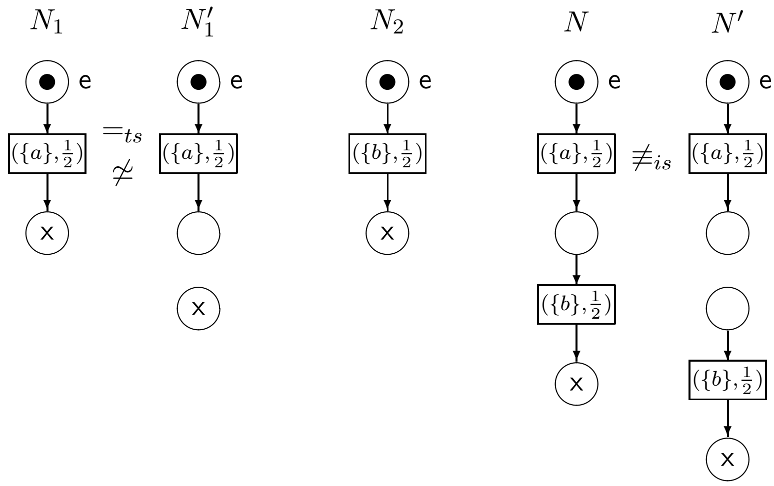

Figure 4. The equivalences between $\equiv_{is}$ and $=_{ts}$ are not congruences

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

In Figure 4 the marked dts-boxes corresponding to the dynamic expressions above are presented, i.e., $N_1 = Box_{dts}(\overline{E})$, $N_1' = Box_{dts}(\overline{E'})$, $N_2 = Box_{dts}(\overline{F})$ and $N = Box_{dts}(\overline{E[]F})$, $N' = Box_{dts}(\overline{E'[]F})$. In addition, we depict the net analogues of the algebraic equivalences.

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

Proposition 5. *The equivalence $\simeq$ is preserved by algebraic operations.*

Proof. By definition of $\simeq$.

We suppose that, for an analogue of $=_{ts}$ to be a congruence, we have to equip transition systems of expressions with two extra transitions skip

and redo like in [8]. This allows one to avoid difficulties demonstrated in the example from the proof of Proposition 4 with unexpected termination due to the Stop process. At the same time, such an enrichment of transition systems does not overcome the problems explained in Example 2 with abstraction from empty loops. Hence, the equivalences between $\equiv_{is}$ and $=_{sto}$ defined on the basis of the enriched transition systems will still be non-congruences.

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

$$\overline{E} \stackrel{\mathsf{skip}}{\to} \underline{E} \qquad \underline{E} \stackrel{\mathsf{redo}}{\to} \overline{E}$$

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

Definition 21. Let E be a static expression and $TS(\overline{E}) = (S, L, \mathcal{T}, s)$. The *(labeled probabilistic) sr-transition system* of $\overline{E}$ is a quadruple $TS_{sr}(\overline{E}) = (S_{sr}, L_{sr}, \mathcal{T}_{sr}, s_{sr})$, where

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

We define a new notion of isomorphism for sr-transition systems.

Definition 22. Let E, E' be static expressions and $TS_{sr}(\overline{E}) = (S_{sr}, L_{sr}, \mathcal{T}_{sr}, s_{sr})$, $TS_{sr}(\overline{E'}) = (S'_{sr}, L'_{sr}, \mathcal{T}'_{sr}, s'_{sr})$ be their sr-transition systems. A mapping $\beta : S_{sr} \to S'_{sr}$ is an *isomorphism* between $TS_{sr}(\overline{E})$ and $TS_{sr}(\overline{E'})$, denoted by $\beta : TS_{sr}(\overline{E}) \simeq TS_{sr}(\overline{E'})$, if

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

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

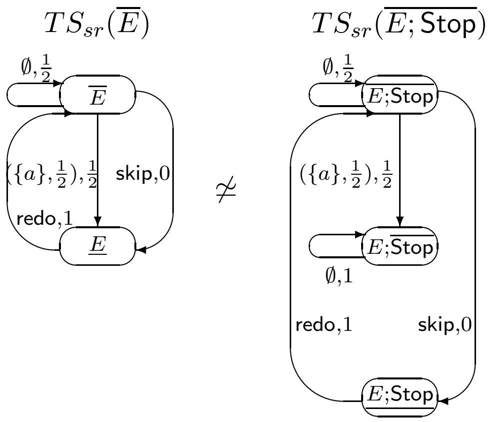

Figure 5. The sr-transition systems of $\overline{E}$ and $\overline{E;\mathsf{Stop}}$ for $E = (\{a\}, \frac{1}{2})$

For $E \in RegStatExpr$, let $TS_{sr}(E) = TS_{sr}(\overline{E})$.

Example 3. Let $E = (\{a\}, \frac{1}{2})$. In Figure 5 the transition systems $TS_{sr}(\overline{E})$ and $TS_{sr}(\overline{E;\mathsf{Stop}})$ are presented. In the latter sr-transition system (unlike the former one) the final state can be reached by executing the transition $(\mathsf{skip}, 0)$ from the initial state only.

Definition 23. Two dynamic expressions $\overline{E}$ and $\overline{E'}$ are *isomorphic with respect to sr-transition systems*, denoted by $\overline{E} =_{tssr} \overline{E'}$, if $TS_{sr}(\overline{E}) \simeq TS_{sr}(\overline{E'})$.

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

Theorem 4. *Let* $\leftrightarrow, \twoheadleftarrow\!\!\!\!\twoheadrightarrow \in \{\equiv, \underline{\leftrightarrow}, =, \simeq\}$ *and* $\star, \star\star \in \{\_, is, ss, sto, ts, tssr\}$. *For dynamic expressions* G *and* G'

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

iff there exists a directed path from $\leftrightarrow_\star$ *to* $\twoheadleftarrow\!\!\!\!\twoheadrightarrow_{\star\star}$ *in the graph in Figure 6.*

Proof. ($\Leftarrow$) Let us check validity of implications in the graph in Figure 6.

- The implication $=_{tssr} \Rightarrow =_{ts}$ is valid, since sr-transition systems have more states and transitions than usual ones.

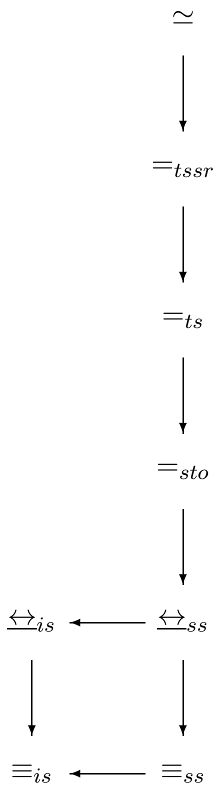

Figure 6. Interrelations of the stochastic equivalences and the new congruence

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

($\Rightarrow$) The absence of additional nontrivial arrows (not resulting from the combination of the existing ones) in the graph in Figure 6 is proved by the following examples.

- Let $E = (\{a\}, \frac{1}{2})$ and $E' = (\{a\}, \frac{1}{2}); \mathsf{Stop}$. We have $\overline{E} =_{ts} \overline{E'}$ as demonstrated in the example from the proof of Proposition 4. On the other hand, $\overline{E} \neq_{tssr} \overline{E'}$, since only in $TS_{sr}(\overline{E'})$ after the transition with the multiaction part of label $\{a\}$ we do not reach the final state, see Example 3.
- Let $E = (\{a\}, \frac{1}{2})$ and $E' = ((\{a\}, \frac{1}{2}); (\{\hat{a}\}, \frac{1}{2}))$ sy a. Then $\overline{E} =_{tssr} \overline{E'}$, since $\overline{E} =_{ts} \overline{E'}$ as demonstrated in the last example from the proof of Theorem 3, and the final states of both $TS_{sr}(\overline{E'})$ and $TS_{sr}(\overline{E'})$ are reachable from the others with "normal" transitions (i.e., not with skip only). On the other hand, $\overline{E} \not\simeq \overline{E'}$.

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

Theorem 5. *Let* $a \in Act$ *and* $E, E', F, K \in RegStatExpr$. *If* $\overline{E} =_{tssr} \overline{E'}$ *then*

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

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

7. Conclusion

In this paper, in the framework of $dtsPBC$ with iteration, we considered a number of stochastic algebraic equivalences which have natural net analogues on LDTSPNs. The equivalences abstract from empty loops in transition systems corresponding to dynamic expressions. Preservation of the equivalences by algebraic operations was investigated. As a result, a congruence relation was proposed based on the transition systems isomorphism.

In the future, we plan to demonstrate how the equivalence notions can be applied to reduction of expressions and boxes. A logical characterization of the equivalences via formulas of probabilistic modal logics is an important problem as well. The next subject is to investigate which equivalence relations preserve stationary behaviour. Further, we intend to outline the performance evaluation technique within the algebra and present the corresponding case studies.

References

[1] Best E., Devillers R., Hall J.G. The box calculus: a new causal algebra with multi-label communication // Lect. Notes Comput. Sci. – 1992. – Vol. 609. – P. 21–69.

[2] Bernardo M., Gorrieri R. A tutorial on EMPA: a theory of concurrent processes with nondeterminism, priorities, probabilities and time // Theor. Comput. Sci. – 1998. – Vol. 202. – P. 1–54.

[3] Hillston J. A compositional approach to performance modelling. – Cambridge University Press, 1996.

[4] Hermanns H., Rettelbach M. Syntax, semantics, equivalences and axioms for MTIPP // Proc. of 2^{nd} Workshop on Process Algebras and Performance Modelling. – 1994. – Vol. 27. – P. 71–88.

[5] Milner R.A.J. Communication and concurrency. – NY: Prentice-Hall International, 1989.

[6] Macià H.S., Valero V.R., Cazorla D.L., Cuartero F.G. Introducing the iteration in sPBC. – Department of Computer Science, University of Castilla-La Mancha, Albacete, Spain, September 2003. – 20 p. – (Technical Report; Vol. DIAB-03-01-37). – `http://www.info-ab.uclm.es/descargas/tecnicalreports/DIAB-03-01-37/diab030137.zip`

[7] Macià H.S., Valero V.R., Cazorla D.L., Cuartero F.G. Introducing the iteration in sPBC // Lect. Notes Comp. Sci. – 2004. – Vol. 3235. – P. 292–308. – `http://www.info-ab.uclm.es/retics/publications/2004/forte04.pdf`

[8] Macià H.S., Valero V.R., Cuartero F.G., de Frutos D.E. A congruence relation for sPBC // Formal Methods in System Design. – Springer, The Netherlands, April 2008. – Vol. 32, N 2. – P. 85–128.

[9] Macià H.S., Valero V.R., de Frutos D.E. sPBC: a Markovian extension of finite Petri box calculus // Proc. of 9^{th} IEEE Internat. Workshop on Petri Nets and Performance Models - 01 (PNPM'01). – Aachen: IEEE Computer Society Press, 2001. – P. 207–216. – `http://www.info-ab.uclm.es/retics/publications/2001/pnpm01.ps`

[10] Macià H.S., Valero V.R., de Frutos D.E., Cuartero F.G. Extending PBC with Markovian multiactions // Proc. of XXVII Conferencia Latinoamericana de Informática - 01 (CLEI'01) (Montilva J.A., Besembel I., eds.). – Mérida, Venezuela, Universidad de los Andes, September 2001. – 12 p. – `http://www.info-ab.uclm.es/retics/publications/2001/clei01.ps`

[11] Tarasyuk I.V. Discrete time stochastic Petri box calculus. – Carl von Ossietzky Universität Oldenburg, 2005. – 25 p. – (Berichte aus dem Department für Informatik; Vol. 3/05). – `http://www.iis.nsk.su/persons/itar/dtspbcib_cov.pdf`

[12] Tarasyuk I.V. Iteration in discrete time stochastic Petri box calculus // Bull. Novosibirsk Comp. Center. Ser. Computer Science. – Novosibirsk, 2006. – Iss. 24. – P. 129–148.
– `http://www.iis.nsk.su/persons/itar/dtsitncc.pdf`

[13] Tarasyuk I.V. Stochastic Petri box calculus with discrete time // Fundamenta Informaticae. – IOS Press, Amsterdam, The Netherlands, February 2007. – Vol. 76, N 1–2. – P. 189–218. – `http://www.iis.nsk.su/persons/itar/dtspbcfi.pdf`