CARL VON OSSIETZKY universität OLDENBURG

BERICHTE

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

Herausgeber: Die Professorinnen und Professoren des Departments für Informatik

Equivalences for modular performance analysis in *dtsPBC*

Igor V. Tarasyuk

Bericht

Nummer 04/11 – Oktober 2011
ISSN 1867-9218

BERICHTE

AUS DEM DEPARTMENT FÜR INFORMATIK

der Fakultät II - Informatik, Wirtschafts- und Rechtswissenschaften

Herausgeber: Die Professorinnen und Professoren
des Departments für Informatik

Equivalences for modular performance analysis in *dtsPBC*

Igor V. Tarasyuk

Bericht

Nummer 04/11 – Oktober 2011

ISSN 1867-9218

Equivalences for modular performance analysis in dtsPBC*

Igor V. Tarasyuk

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

Abstract

In the framework of a discrete time stochastic extension dtsPBC of finite Petri box calculus (PBC) enriched with iteration, we define a number of stochastic equivalences. They allow one to identify stochastic processes with similar behaviour that are however differentiated by the semantics of the calculus. We explain in which way the equivalences we propose can be used to reduce transition systems of expressions. It is demonstrated how to apply the equivalences to compare the stationary behaviour. The equivalences guarantee a coincidence of performance indices for stochastic systems and can be used for performance analysis simplification. In a case study, a method of modeling, performance evaluation and behaviour preserving reduction of concurrent computing systems is outlined and applied to the dining philosophers system.

Keywords: stochastic process algebra, Petri box calculus, iteration, discrete time, stochastic equivalence, reduction, stationary behaviour, performance evaluation.

1 Introduction

Process algebras (PAs), such as CCS [23], are a widely used formal model designed to specify concurrent systems and analyze their behavioural properties. In such calculi, processes are specified by compositional formulas constructed by operators from symbols of actions, and verification of properties is accomplished syntactically by means of algebraic laws and equivalences. In the last decades, stochastic extensions of PAs were proposed. Stochastic process algebras (SPAs) do not just specify actions which can occur (qualitative features), like standard PAs, but they associate with actions the distribution parameters of their random time delays (quantitative characteristics). The most well-known SPAs are MTIPP [16], PEPA [15] and EMPA [6].

Petri box calculus (PBC) [3, 9] is a flexible and expressive process algebra intended to provide support for compositional translation from high level concurrent programming languages into Petri nets (PNs). Formulas of PBC are combined not from single actions, like in CCS, but from multisets of elementary actions and their conjugates, called multiactions (*basic formulas*). In contrast to CCS, synchronization is separated from parallelism (*concurrent constructs*). Synchronization is defined as a unary multi-way stepwise operation based on communication of actions and their conjugates, thus, it extends the CCS approach with conjugate matching labels. Synchronization in PBC is asynchronous, unlike that in Synchronous CCS (SCCS) [23]. The other operations are sequence and choice (*sequential constructs*). The calculus includes also restriction and relabeling (*abstraction constructs*). To specify infinite processes, the refinement, recursion and iteration operations were added (*hierarchical constructs*). Thus, unlike CCS, PBC has an additional iteration construction to specify infiniteness when the semantic interpretation in finite PNs is possible. PBC has a step operational semantics in terms of labeled transition systems based on rules in the Structured Operational Semantics (SOS) style. A denotational semantics of PBC was proposed via a subclass of PNs equipped with an interface and considered up to isomorphism, called Petri boxes. Recently, the extensions of PBC with deterministic or stochastic time were presented.

A deterministic time model is considered in time Petri box calculus (tPBC) [20], in timed Petri box calculus (TPBC) [22] and in arc time Petri box calculus (atPBC) [31]. In tPBC, timing information is added with combining instantaneous multiactions and time delays. Its denotational semantics was defined in terms of

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

a subclass of labeled time PNs (LtPNs), called time Petri boxes (ct-boxes). tPBC has an interleaving time operational semantics in terms of labeled transition systems. In contrast to tPBC, multiactions of TPBC are not instantaneous but have time durations. The denotational semantics of TPBC was defined via a subclass of labeled timed PNs (LTPNs), called timed Petri boxes (T-boxes). TPBC has a step timed operational semantics in terms of labeled transition systems. In atPBC, multiactions are associated with time delay intervals, and a step operational semantics is defined. The denotational semantics is defined on a subclass of arc time PNs (atPNs), where time restrictions are associated with the arcs, called arc time Petri boxes (at-boxes).

A continuous time stochastic extension of a finite part of PBC called stochastic Petri box calculus (sPBC) was proposed in [28]. sPBC in its former version had neither refinement nor recursion nor iteration operations and thus specified finite processes only. An interleaving operational semantics of the calculus was constructed in terms of labeled probabilistic transition systems. A denotational semantics of sPBC was defined via a subclass of labeled continuous time stochastic PNs (LCTSPNs) called stochastic Petri boxes (s-boxes). In [26], the iteration operation was added to sPBC to specify infinite processes and the producer/consumer system was specified. In [27], the resulting calculus was enriched with immediate multiactions, and a manufacturing system, as well as the AUY-protocol, were modeled. The example systems considered within sPBC and its extensions had an interleaving semantics and no performance indices were calculated for many of them.

A discrete time stochastic extension dtsPBC of finite PBC was presented in [35, 37]. A step operational semantics of the algebra was constructed with the use of labeled probabilistic transition systems. dtsPBC has a denotational semantics based on a subclass of labeled discrete time stochastic PNs (LDTSPNs) called discrete time stochastic Petri boxes (dts-boxes). A number of stochastic equivalences were proposed to identify stochastic processes with similar behaviour which are differentiated by the semantic equivalence. The interrelations of the introduced equivalences were studied. In [36, 38], the syntax of dtsPBC was supplemented by the iteration operator. In [39], we presented the extension dtsiPBC of the latter calculus with immediate multiactions.

Since dtsPBC has a discrete time semantics and geometrically distributed delays in the process states, unlike sPBC with continuous time semantics and exponentially distributed delays, the calculi apply two different approaches to the stochastic extension of PBC, in spite of some similarity of their syntax and semantics inherited from PBC. The main advantage of dtsPBC is that concurrency is treated naturally, like in PBC, whereas in sPBC parallelism is simulated by interleaving, obliging one to collect the information on causal independence of activities before constructing the semantics.

If to compare dtsPBC with classical SPAs MTIPP, PEPA, EMPA, the first main difference between them comes from PBC, since dtsPBC is based on this calculus: all algebraic operations and a notion of multiaction are inherited from PBC. The second main difference is discrete conditional probabilities of activities induced by the discrete time approach, whereas action rates are used in the standard SPAs with continuous time. As a consequence, dtsPBC has a non-interleaving step operational semantics, in contrast to the classical SPAs, where concurrency is modeled by interleaving.

A notion of equivalence is important in theory of computing systems. Equivalences are applied both to compare behaviour of systems and reduce their structure. There is a wide diversity of behavioural equivalences, and their interrelations were well explored in the literature. The most well-known and widely used one is bisimulation. Standardly, the mentioned equivalences take into account only functional (qualitative) but not performance (quantitative) aspects. Additionally, the equivalences are usually interleaving ones, i.e., they interpret concurrency as sequential nondeterminism. To respect quantitative features of behaviour, equivalences for SPAs have additional requirement on execution probabilities. Two equivalent processes must be able to execute the same sequences of actions, and for every such sequence, its execution probabilities within both processes should coincide. In case of bisimulation equivalence, the states from which similar future behaviours start are grouped into equivalence classes that form elements of the aggregated state space. From every two bisimilar states, the same actions can be executed, and the subsequent states resulting from execution of an action belong to the same equivalence class. In addition, for both states, the cumulative probabilities to move to the same equivalence class by executing the same action coincide.

Interleaving probabilistic strong bisimulation equivalence was proposed in [21] on labeled probabilistic transition systems, in [16] on labeled CTMCs and in [15] on probabilistic process algebras. Interleaving probabilistic equivalences were defined for probabilistic processes in [14, 17]. Interleaving probabilistic weak bisimulation equivalence was introduced in [11] on labeled CTSPNs and in [12] on generalized SPNs (GSPNs). In [5], a comparison of a variety of interleaving Markovian trace, test and bisimulation equivalences was carried out on sequential and concurrent Markovian process calculi. At the same time, no appropriate equivalence notion was defined for concurrent SPAs so far.

In this paper, a problem of performance preservation by the equivalence notions is discussed within dtsPBC enriched with iteration. First, we present the syntax of the calculus. Second, we describe its operational semantics in terms of labeled transition systems and its denotational semantics based on a subclass of LDTSPNs. Further, we propose a number of stochastic equivalences. We describe how the stochastic equivalences can be used to reduce transition systems of expressions and the related formalisms while preserving their qualitative and quantitative behaviour. We investigate which equivalences guarantee identity of the stationary behaviour. The mentioned property implies a coincidence of performance indices based on steady-state probabilities of modeled stochastic systems. The equivalences possessing the property can be used to reduce the state space of a system and thus simplify its performance evaluation, that is usually complex due to the state space explosion problem. At the end, we present a case study of the dining philosophers system explaining how to model concurrent computing systems within the calculus and analyze their performance, as well as in which way to reduce the systems preserving their performance indices and making simpler the performance evaluation.

The paper is organized as follows. The syntax of dtsPBC is presented in Section 2. Section 3 describes the operational semantics of the calculus and Section 4 presents its denotational semantics. Stochastic algebraic equivalences are defined and investigated in Section 5. In Section 6 we explain how to reduce transition systems and the related formalisms modulo the equivalences. Section 7 is devoted to the application of the relations to the stationary behaviour comparison and determining the performance preserving equivalences. Section 8 describes specification, performance evaluation and reduction of the dining philosophers system within the calculus. The difference between dtsPBC and other well-known or similar SPAs is discussed in Section 9. The concluding Section 10 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, *discrete time stochastic* PBC (dtsPBC).

We denote the *set of all finite multisets* over X by $\mathbb{N}_f^X$. 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 $\hat{a} \neq a$ and $\hat{\hat{a}} = a$. Let $\mathcal{A} = Act \cup \widehat{Act}$ be the set of *all actions*, and $\mathcal{L} = \mathbb{N}_f^{\mathcal{A}}$ 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. 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 $\mathcal{A}(\alpha, \rho) = \mathcal{A}(\alpha)$. The *alphabet* of $\Gamma \in \mathbb{N}_f^{\mathcal{SL}}$ is defined as $\mathcal{A}(\Gamma) = \cup_{(\alpha,\rho)\in\Gamma}\mathcal{A}(\alpha)$. For $(\alpha, \rho) \in \mathcal{SL}$, we define its *multiaction part* as $\mathcal{L}(\alpha, \rho) = \alpha$ and its *probability part* as $\Omega(\alpha, \rho) = \rho$. The *multiaction part* of $\Gamma \in \mathbb{N}_f^{\mathcal{SL}}$ is defined as $\mathcal{L}(\Gamma) = \sum_{(\alpha,\rho)\in\Gamma} \alpha$. Remember that sums are considered with the multiplicity when applied to multisets.

Activities are combined into formulas by the following operations: *sequential execution* ;, *choice* [], *parallelism* ‖, *relabeling* $[f]$ of actions, *restriction* rs over a single action, *synchronization* sy on an action and its conjugate, and *iteration* [∗∗] with three arguments: initialization, body and termination.

Sequential execution and choice have standard interpretation, like in other process algebras, but parallelism does not include synchronization, unlike the corresponding operation in *CCS*.

Relabeling functions $f : \mathcal{A} \to \mathcal{A}$ are bijections preserving conjugates, i.e., $\forall x \in \mathcal{A}\ f(\hat{x}) = \widehat{f(x)}$. Relabeling is extended to multiactions in a usual way: for $\alpha \in \mathcal{L}$ we define $f(\alpha) = \sum_{x\in\alpha} f(x)$. Relabeling is extended to the multisets of activities as follows: for $\Gamma \in \mathbb{N}_f^{\mathcal{SL}}$ we define $f(\Gamma) = \sum_{(\alpha,\rho)\in\Gamma}(f(\alpha), \rho)$.

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

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

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

In the iteration, the initialization subprocess is executed first, then the body is performed zero or more times and, finally, the termination is executed.

Static expressions specify the structure of processes. The expressions correspond to unmarked LDTSPNs (note that LDTSPNs are marked by definition).

Definition 2.1 *Let* $(\alpha, \rho) \in \mathcal{SL},\ a \in Act$. *A* static expression *of dtsPBC is*

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

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

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

To avoid inconsistency of the iteration operator, we should not allow any concurrency at the highest level of the second argument of iteration. This is not a severe restriction though, since we can always prefix parallel expressions by an activity with the empty multiaction. Later on, in Example 4.3, we shall demonstrate that such inconsistency can result in nets which are not safe, see also [4] for discussion on this subject.

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

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

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

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

Definition 2.3 *Let* $E \in StatExpr$, $a \in Act$. *A* dynamic expression *of dtsPBC is*

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

DynExpr denotes the set of *all dynamic expressions* of dtsPBC.

If the underlying static expression of a dynamic one is not regular, then the corresponding LDTSPN can be non-safe, as Example 4.3 will show (but the LDTSPN is 2-bounded in the worst case, see [4]). A dynamic expression is *regular* if its underlying static expression is regular. *RegDynExpr* denotes the set of *all regular dynamic expressions* of dtsPBC.

3 Operational semantics

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

3.1 Inaction rules

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

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

In Table 1, we define inaction rules for regular dynamic expressions in the form of overlined and underlined regular static ones. In this table, $E, F, K \in RegStatExpr$, $a \in Act$.

In Table 2, we propose inaction rules for regular dynamic expressions in the arbitrary form. In this table, $E, F \in RegStatExpr$, $G, H, \widetilde{G}, \widetilde{H} \in RegDynExpr$ and $a \in Act$.

A regular dynamic expression G is *operative* if no inaction rule can be applied to it. *OpRegDynExpr* denotes the set of *all operative regular dynamic expressions* of dtsPBC. Any regular dynamic expression can be transformed into a (possibly not unique) operative one by the inaction rules. In the following, we consider regular expressions only and omit the word “regular”.

Definition 3.1 *Let* $\approx\ = (\Rightarrow \cup \Leftarrow)^*$ *be a structural equivalence of dynamic expressions in dtsPBC. Thus, two dynamic expressions* G *and* G' *are* structurally equivalent, *denoted by* $G \approx G'$, *if they can be reached from each other by applying the inaction rules in a forward or backward direction.*

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

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

Table 2: Inaction rules for arbitrary regular dynamic expressions

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

3.2 Action and empty loop rules

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

The action rules describe dynamic expression transformations due to execution of non-empty multisets of activities. The rules represent possible state changes of the specified processes when some non-empty multisets of activities are executed. As we shall see, the application of an action rule to a dynamic expression leads to a discrete time step in the corresponding LDTSPN at which some transitions are fired and the current marking is changed, unless there is a self-loop produced by the iterative execution of a non-empty multiset (which, additionally, should be one-element, i.e., the single activity, since we do not allow concurrency at the highest level of the second argument of iteration).

The empty loop rule $G \stackrel{\emptyset}{\to} G$ describes dynamic expression transformations due to execution of the empty multiset of activities at a discrete time step. The rule reflects a non-zero probability to stay in the current state at the next time moment, which is an essential feature of discrete time stochastic processes. As we shall see, the application of the empty loop rule to a dynamic expression leads to a discrete time step in the corresponding LDTSPN at which no transitions are fired and the current marking is not changed. This is a new rule that has no prototype among inaction rules of PBC, since it represents a time delay. The PBC rule $G \stackrel{\emptyset}{\to} G$ from [4] in our setting would correspond to the rule $G \Rightarrow G$ describing the stay in the current state when no time elapses, but it is not needed to transform dynamic expressions into operative ones.

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

In Table 3, we define the action and empty loop rules. In this table, (α, ρ), $(\beta, \chi) \in \mathcal{SL}$, $E, F \in$ *RegStatExpr*, $G, H \in$ *OpRegDynExpr*, $\widetilde{G}, \widetilde{H} \in$ *RegDynExpr* and $a \in$ *Act*. Moreover, $\Gamma, \Delta \in I\!N_f^{\mathcal{SL}} \setminus \{\emptyset\}$ and $\Gamma' \in I\!N_f^{\mathcal{SL}}$.

Rule **Sy2** establishes that the synchronization of two multiactions is made by taking the product of their probabilities, since we are considering that both must occur for the synchronization to happen, so this corresponds, in some sense, to the probability of the independent events intersection, but the real situation is more complex, since these multiactions can be also executed in parallel. Instead of multiplication, we can use any associative and commutative binary operation $\odot : (0;1)^2 \to (0;1)$. In addition, for the probabilities ρ and χ of two synchronized multiactions we require that $\rho \odot \chi \leq \min\{\rho, \chi\}$ to express an idea that the probability of the resulting synchronized multiaction should not be greater than those of the two synchronized ones, since, while

Table 3: Action and empty loop rules

El $G \stackrel{\emptyset}{\rightarrow} G$	**B** $\overline{(\alpha,\rho)} \xrightarrow{\{(\alpha,\rho)\}} \underline{(\alpha,\rho)}$	**SC1** $\frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G},\ \circ\in\{;,[]\}}{G\circ E \stackrel{\Gamma}{\rightarrow} \widetilde{G}\circ E}$	**SC2** $\frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G},\ \circ\in\{;,[]\}}{E\circ G \stackrel{\Gamma}{\rightarrow} E\circ \widetilde{G}}$
P1 $\frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G}}{G\|H \stackrel{\Gamma}{\rightarrow} \widetilde{G}\|H}$	**P2** $\frac{H \stackrel{\Gamma}{\rightarrow} \widetilde{H}}{G\|H \stackrel{\Gamma}{\rightarrow} G\|\widetilde{H}}$	**P3** $\frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G},\ H \stackrel{\Delta}{\rightarrow} \widetilde{H}}{G\|H \xrightarrow{\Gamma+\Delta} \widetilde{G}\|\widetilde{H}}$	**L** $\frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G}}{G[f] \xrightarrow{f(\Gamma)} \widetilde{G}[f]}$
Rs $\frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G},\ a,\hat{a}\not\in\mathcal{A}(\Gamma)}{G \text{ rs } a \stackrel{\Gamma}{\rightarrow} \widetilde{G} \text{ rs } a}$	**I1** $\frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G}}{[G{*}E{*}F] \stackrel{\Gamma}{\rightarrow} [\widetilde{G}{*}E{*}F]}$	**I2** $\frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G}}{[E{*}G{*}F] \stackrel{\Gamma}{\rightarrow} [E{*}\widetilde{G}{*}F]}$	**I3** $\frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G}}{[E{*}F{*}G] \stackrel{\Gamma}{\rightarrow} [E{*}F{*}\widetilde{G}]}$
Sy1 $\frac{G \stackrel{\Gamma}{\rightarrow} \widetilde{G}}{G \text{ sy } a \stackrel{\Gamma}{\rightarrow} \widetilde{G} \text{ sy } a}$	**Sy2** $\frac{G \text{ sy } a \xrightarrow{\Gamma'+\{(\alpha,\rho)\}+\{(\beta,\chi)\}} \widetilde{G} \text{ sy } a,\ a\in\alpha,\ \hat{a}\in\beta}{G \text{ sy } a \xrightarrow{\Gamma'+\{(\alpha\oplus_a\beta,\rho\cdot\chi)\}} \widetilde{G} \text{ sy } a}$		

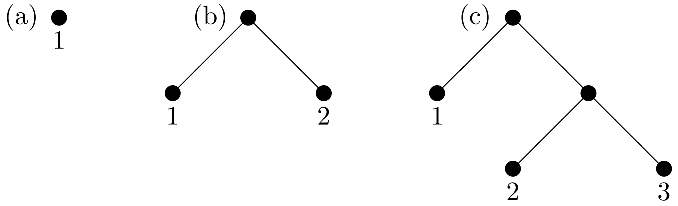

Figure 1: The binary trees encoded with the numberings 1, (1)(2) and (1)((2)(3))

performance evaluation, it is usually supposed that the execution of two components together require more system resources than the execution of each single one. It is clear that the multiplication operation satisfies the mentioned properties and it is easy to handle with, since it naturally maintains compositionality, thus, we have chosen the product of the probabilities. As we shall see, for every LDTSPN obtained by synchronization of two LDTSPNs, this approach allows us to calculate the transition firing probabilities using the standard transition probability function for that net class. See also [8, 10] for a discussion about binary operations producing the rates of synchronization in the continuous time setting.

We do not allow self-synchronization, i.e., synchronization of an activity with itself, to avoid an unexpected behaviour and many technical difficulties, see [4].

3.3 Transition systems

Now we construct labeled probabilistic transition systems associated with dynamic expressions and used to define the operational semantics of dtsPBC.

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

Definition 3.2 *Let $n \in I\!N$. The* numbering *of expressions is defined as* $\iota ::= \ n \mid (\iota)(\iota)$.

Num denotes the set of *all numberings* of expressions.

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

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

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

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

Definition 3.3 *Let G be a dynamic expression. Then $[G]_{\approx} = \{H \mid G \approx H\}$ is the equivalence class of G w.r.t. the structural equivalence. The* derivation set *of a dynamic expression G, denoted by $DR(G)$, is the minimal set such that*

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

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

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

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

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

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

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

Thus, if $Exec(s) \neq \{\emptyset\}$, then $PF(\Gamma, s)$ could be interpreted as a *joint* probability of independent events. Each such an event is interpreted as readiness or not readiness for execution of a particular activity from Γ. Every activity belonging to some multiset from $Exec(s)$ decides probabilistically (using its probabilistic part) and independently (from other activities), if it is ready for execution in s. The multiplication in the definition is used because it reflects the probability of the independent events intersection. When only the empty multiset of activities can be executed in s, i.e., $Exec(s) = \{\emptyset\}$, we have $PF(\emptyset, s) = 1$, since we stay in s in this case.

Let $\Gamma \in Exec(s)$. The *probability to execute the multiset of activities Γ in s* is

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

Thus, $PT(\Gamma, s)$ is the probability that Γ is ready for execution in s *normalized* by the analogous probability for *any* multiset executable in s. The denominator of the fraction above is a sum since it reflects the probability of the mutually exclusive events union. $PT(\Gamma, s)$ can be interpreted as the *conditional* probability to execute Γ in s under condition that only *some particular* multisets (including the empty one) of activities belonging to multisets from $Exec(s)$ can be executed in s. If $Exec(s) \neq \{\emptyset\}$, then $PF(\Gamma, s)$ can be seen as the *potential* probability to execute Γ in s, since we have $PF(\Gamma, s) = PT(\Gamma, s)$ only when *all* finite multisets (including the empty one) of activities belonging to multisets from $Exec(s)$ can be executed in s. In this case, all the mentioned activities can be executed in parallel in s and we have $\sum_{\Delta\in Exec(s)} PF(\Delta, s) = 1$, since this sum collects the products of *all* combinations of the probability parts of the activities and the negations of these parts.

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

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

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

Definition 3.4 *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 $L_G \subseteq \mathbb{N}_f^{\mathcal{SL}} \times (0;1]$;*
- *the set of* transitions *is $\mathcal{T}_G = \{(s, (\Gamma, PT(\Gamma, s)), \tilde{s}) \mid s \in DR(G),\ \exists H \in s\ \exists \widetilde{H} \in \tilde{s}\ H \stackrel{\Gamma}{\to} \widetilde{H}\}$;*
- *the* initial state *is $s_G = [G]_{\approx}$.*

The definition of $TS(G)$ is correct, i.e., for every state, the sum of the probabilities of all the transitions starting from it is 1. This is guaranteed by the note after the definition of $PT(\Gamma, s)$. Thus, we have defined a *generative* model of probabilistic processes [17], according to the classification from [14]. The reason is that the sum of the probabilities of the transitions with all possible labels should be equal to 1, not only of those with the same labels (up to enumeration of activities they include) as in the *reactive* models [21], and we do not have the nested probabilistic choice as in the *stratified* models [14].

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

Note that Γ can be the empty multiset, and its execution does not change the current state (the equivalence class), since we have a loop transition $s \stackrel{\emptyset}{\rightarrow}_\mathcal{P} s$ from a state s to itself in the result of executing the empty multiset. This corresponds to application of the empty loop rule to expressions from the equivalence class. We have to keep track of such executions, called *empty loops*, because they have nonzero probabilities. This follows from the definition of $PF(\emptyset, s)$ and the fact that multiaction probabilities cannot be equal to 1 as they belong to the interval $(0; 1)$. The step probabilities belong to the interval $(0; 1]$. The step probability is 1 when we cannot leave a state s, hence, there exists only one transition from it, namely, the empty loop transition $s \stackrel{\emptyset}{\rightarrow}_1 s$.

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

Isomorphism is a coincidence of systems up to renaming of their components. Let $\simeq$ denote isomorphism between transition systems that binds their initial states.

Definition 3.5 *Dynamic expressions G and G' are* equivalent w.r.t. 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 the transitions $s \rightarrow_\mathcal{P} \tilde{s}$, if $s \rightarrow \tilde{s}$ and $\mathcal{P} = PM(s, \tilde{s})$.*

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

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

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

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

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

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

Example 3.2 *Let $E_1 = (\{a\}, \rho)[](\{a\}, \rho),\ E_2 = (\{b\}, \chi),\ E_3 = (\{c\}, \theta)$ and $E = [E_1 * E_2 * E_3]$. The identical activities of the composite static expression are enumerated as follows: $E = [((\{a\}, \rho)_1[](\{a\}, \rho)_2) * (\{b\}, \chi) * (\{c\}, \theta)]$. In Figure 2, the transition system $TS(\overline{E})$ and the underlying DTMC $DTMC(\overline{E})$ are presented. For simplicity, the states are labeled by expressions belonging to the corresponding equivalence classes, and singleton multisets of activities are written without braces. $DR(\overline{E})$ consists of the equivalence classes $s_1 = [[\overline{E_1 * E_2 * E_3}]]_\approx,\ s_2 = [[E_1 * \overline{E_2} * E_3]]_\approx,\ s_3 = [\underline{[E_1 * E_2 * E_3]}]_\approx$. The average sojourn time vector of $\overline{E}$ is $SJ = \left(\frac{1+\rho}{2\rho}, \frac{1-\chi\theta}{\theta(1-\chi)}, \infty\right)$. The sojourn time variance vector of $\overline{E}$ is $VAR = \left(\frac{(1+\rho)^2}{4\rho^2}, \frac{(1-\chi\theta)^2}{\theta^2(1-\chi)^2}, \infty\right)$.*

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

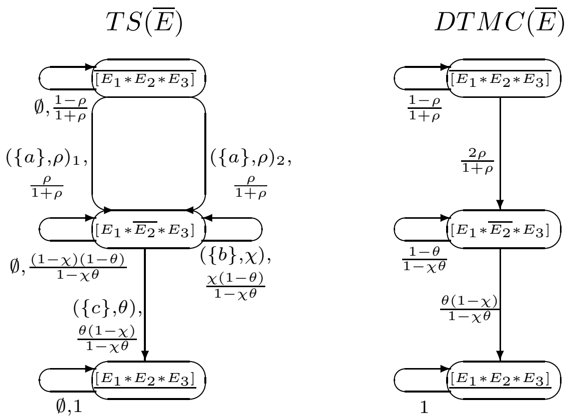

Figure 2: The transition system and the underlying DTMC of $\overline{E}$ for $E = [((\{a\},\rho)_1[](\{a\},\rho)_2)*(\{b\},\chi)*(\{c\},\theta)]$

4 Denotational semantics

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

4.1 Labeled DTSPNs

We introduce a class of labeled discrete time stochastic PNs (LDTSPNs), a subclass of DTSPNs [24] (we do not allow the transition probabilities to be equal to 1) with the transition labeling.

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 providing the* weights of arcs *between places and transitions;*
- $\Omega_N : T_N \to (0;1)$ *is the* transition probability *function associating transitions with probabilities;*
- $L_N : T_N \to \mathcal{L}$ *is the* transition labeling *function assigning multiactions to transitions;*
- $M_N \in I\!N_f^{P_N}$ *is the* initial marking.

A graphical representation of LDTSPNs is, like that for standard labeled PNs, but with probabilities written near the corresponding transitions. In the case the probabilities are not given in the picture, they are considered to be of no importance. The weights of arcs 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 the 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. Moreover, we do not need to consider self-concurrency, since denotational semantics of expressions will be defined via dts-boxes which are safe LDTSPNs (hence, no self-concurrency is possible).

Let M be a marking of an LDTSPN N. A transition $t \in Ena(M)$ fires with probability $\Omega_N(t)$ when no other transitions conflicting with it are enabled.

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

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

In the case $U = \emptyset$ we define

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

Thus, if $Ena(M) \neq \emptyset$, then $PF(U, M)$ could be interpreted as a *joint* probability of independent events. Each such an event is interpreted as readiness or not readiness for firing of a particular transition from U. Every transition from $Ena(M)$ decides probabilistically (using its probability) and independently (from other transitions), if it is ready for firing in M. The multiplication in the definition is used because it reflects the probability of the independent events intersection. When no transitions are enabled in M, i.e. $Ena(M) = \emptyset$, we have $PF(\emptyset, M) = 1$, since we stay in M in this case.

Let $U \subseteq Ena(M)$, $U \neq \emptyset$ and ${}^\bullet U \subseteq M$. The concurrent firing of the transitions from U changes the marking M to $\widetilde{M} = M - {}^\bullet U + U^\bullet$, denoted by $M \xrightarrow{U}_{\mathcal{P}} \widetilde{M}$, where $\mathcal{P} = PT(U, M)$ is the *probability that the set of transitions U fires in M* defined as

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

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 is ready for firing in M *normalized* by the corresponding probability for *any* set enabled in M. The denominator of the fraction above is a sum since it reflects the probability of the mutually exclusive events union. $PT(U, M)$ can be interpreted as the *conditional* probability that U fires in M under condition that only *some particular* subsets (including the empty one) of transitions from $Ena(M)$ can fire in M. If $Ena(M) \neq \emptyset$, then $PF(U, M)$ can be seen as the *potential* probability that U fires in M, since we have $PF(U, M) = PT(U, M)$ only when *all* subsets (including the empty one) of transitions from $Ena(M)$ can fire in M. In this case, all the mentioned transitions can be fired in parallel in M (i.e., $Ena(M)$ can fire in M) and we have $\sum_{\{V \mid {}^\bullet V \subseteq M\}} PF(V, M) = 1$, since this sum collects the products of *all* combinations of the transition probabilities and their negations.

Let $Ena(M) = \{t_1, \ldots, t_n\}$ be a mutually exclusive set of transitions (i.e., firing of any transition from the set results in a marking in which no other transition from the set is enabled) and $\rho_i = \Omega_N(t_i)$ $(1 \leq i \leq n)$. Then $PT(\{t_i\}, M)$ resembles the probabilistic function $P[E_i]$ from [24], which defines the probability of the event E_i, that transition t_i in a mutually exclusive set of transitions $\{t_1, \ldots, t_n\}$ will fire in the marking M. We have $P[E_i] = \frac{\frac{\rho_i}{1-\rho_i}}{1+\sum_{j=1}^{n} \frac{\rho_j}{1-\rho_j}} = \frac{\frac{\rho_i(1-\rho_1)\cdots(1-\rho_n)}{1-\rho_i}}{(1-\rho_1)\cdots(1-\rho_n)+\sum_{j=1}^{n} \frac{\rho_j(1-\rho_1)\cdots(1-\rho_n)}{1-\rho_j}}$, where $\frac{\rho_i(1-\rho_1)\cdots(1-\rho_n)}{1-\rho_i}$ corresponds to $PF(\{t_i\}, M)$ in our setting. Further, $PT(\emptyset, M)$ resembles the probabilistic function $P[E_0]$, which defines the probability of the event E_0, that no transitions from the mutually exclusive set of transitions $\{t_1, \ldots, t_n\}$ will fire in the marking M. We have $P[E_0] = \frac{1}{1+\sum_{j=1}^{n} \frac{\rho_j}{1-\rho_j}} = \frac{(1-\rho_1)\cdots(1-\rho_n)}{(1-\rho_1)\cdots(1-\rho_n)+\sum_{j=1}^{n} \frac{\rho_j(1-\rho_1)\cdots(1-\rho_n)}{1-\rho_j}}$, where $(1 - \rho_1) \cdots (1 - \rho_n)$ corresponds to $PF(\emptyset, M)$ in our setting. If $Ena(M)$ is not a mutually exclusive set of transitions, our way to define $PT(U, M)$ for $U \subseteq Ena(M)$, $U \neq \emptyset$, extends the approach of [24]. The advantage of our two-stage definition of $PT(U, M)$ is that it has a closed form and we do not need to consider which sets of transitions are exclusive, instead, we just consider the probability that U fires in M under condition that only particular subsets of $Ena(M)$ can fire in M.

Note that for all markings of an LDTSPN N the sum of outgoing probabilities is equal to 1, i.e, $\forall M \in \mathbb{N}_f^{P_N}\ PT(\emptyset, M) + \sum_{\{U \mid {}^\bullet U \subseteq M\}} PT(U, M) = 1$. This follows from the definition of $PT(U, M)$ and guarantees that it defines a probability distribution.

We write $M \xrightarrow{U} \widetilde{M}$ if $\exists \mathcal{P}\ M \xrightarrow{U}_{\mathcal{P}} \widetilde{M}$ and $M \to \widetilde{M}$ if $\exists U\ M \xrightarrow{U} \widetilde{M}$.

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*

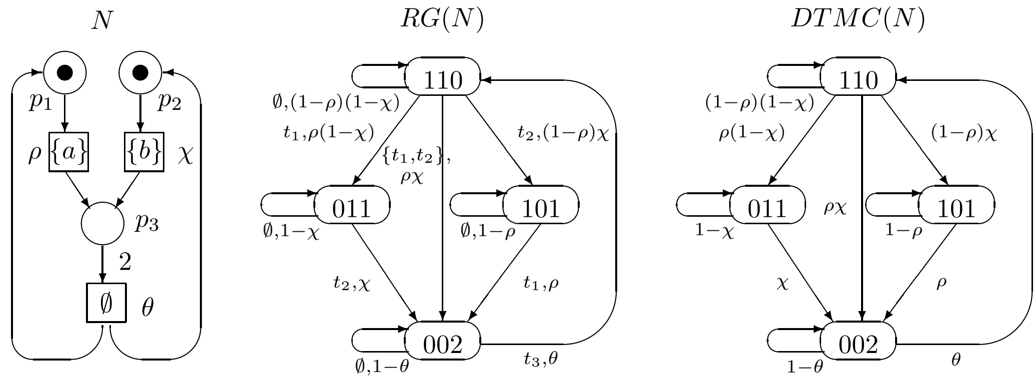

Figure 3: LDTSPN, its reachability graph and the underlying DTMC

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

- *The* reachability graph *of N, denoted by $RG(N)$, is a directed labeled graph with the set of nodes $RS(N)$ and an arc labeled with $(U, \mathcal{P})$ between nodes M and $\widetilde{M}$ if $M \xrightarrow{U}_{\mathcal{P}} \widetilde{M}$.*
- *The* underlying discrete time Markov chain (DTMC) *of N, denoted by $DTMC(N)$, has the state space $RS(N)$ and the transitions $M \to_{\mathcal{P}} \widetilde{M}$, if $M \to \widetilde{M}$, where $\mathcal{P} = PM(M, \widetilde{M})$ is the* probability to move from M to $\widetilde{M}$ by firing any set of transitions *defined as*

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

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

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

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

The *sojourn time variance in the marking M* is

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

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

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

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

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

- $W_N : (P_N \times T_N) \cup (T_N \times P_N) \to \mathbb{N}$ *is a function providing the* weights of arcs *between places and transitions;*

- Λ_N *is the* place and transition labeling *function such that*

  - $\Lambda_N|_{P_N} : P_N \to \{\mathsf{e}, \mathsf{i}, \mathsf{x}\}$ *(it specifies* entry, internal *and* exit *places, respectively);*
  - $\Lambda_N|_{T_N} : T_N \to \{\varrho \mid \varrho \subseteq \mathbb{N}_f^{\mathcal{SL}} \times \mathcal{SL}\}$ *(it associates transitions with the* relabeling relations *on activities).*

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

A dts-box is *plain* if $\forall t \in T_N\ \Lambda_N(t) \in \mathcal{SL}$, i.e., $\Lambda_N(t)$ is a constant relabeling that will be defined later. In case of the constant relabeling, the shorthand notation (by an activity) for $\Lambda_N(t)$ will be used. A *marked plain dts-box* is a pair (N, M_N), where N is a plain dts-box and $M_N \in \mathbb{N}_f^{P_N}$ is its marking. We use the following notation: $\overline{N} = (N, {}^\circ N)$ 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 functions Ω_N and L_N are defined as follows: $\forall t \in T_N\ \Omega_N(t) = \Omega(\Lambda_N(t))$ and $L_N(t) = \mathcal{L}(\Lambda_N(t))$. Behaviour of the marked dts-boxes follows from the firing rule of LDTSPNs. A plain dtsi-box N is n*-bounded* ($n \in \mathbb{N}$) if $\overline{N}$ is so, i.e., $\forall M \in RS(\overline{N})\ \forall p \in P_N\ M(p) \leq n$, and it is *safe* if it is 1-bounded. A plain dtsi-box N is *clean* if $\forall M \in RS(\overline{N})\ {}^\circ N \subseteq M \ \Rightarrow\ M = {}^\circ N$ and $N^\circ \subseteq M \ \Rightarrow\ M = N^\circ$, if there are tokens in all its entry (exit) places then no other places have tokens.

To define a semantic function associating a plain dts-box with every static expression of dtsPBC, we need to propose the *enumeration* function $Enu : T_N \to Num$. It associates numberings with transitions of the plain dts-box N according to those of activities. In the case of synchronization, the function associates concatenation of the parenthesized numberings of the synchronized transitions with a resulting new transition.

The structure of the plain dts-box corresponding to a static expression is constructed, like in PBC, see [4, 9], i.e., we use a simultaneous refinement and relabeling meta-operator (net refinement) in addition to the *operator dts-boxes* corresponding to the algebraic operations of dtsPBC and featuring transformational transition relabelings. Thus, the resulting plain dts-boxes are safe and clean. In the definition of the denotational semantics, we shall apply standard constructions used for PBC. Let Θ denote an *operator box* and u denote a *transition name* from PBC setting.

The relabeling relations $\varrho \subseteq \mathbb{N}_f^{\mathcal{SL}} \times \mathcal{SL}$ 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;

- $\varrho_{(\alpha,\rho)} = \{(\emptyset, (\alpha, \rho))\}$ is the *constant relabeling* identified with $(\alpha, \rho) \in \mathcal{SL}$ itself;

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

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

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

The plain and operator dts-boxes are presented in Figure 4. The symbol i is usually omitted.

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

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

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

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

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

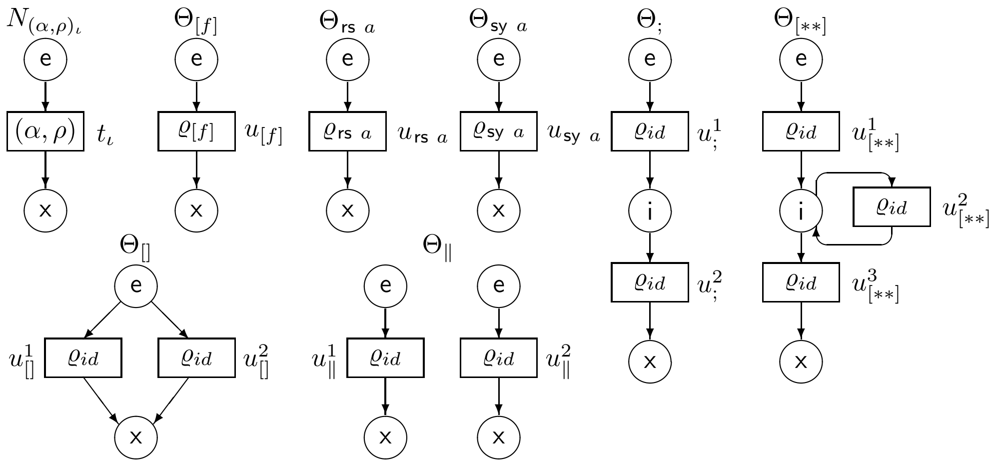

Figure 4: The plain and operator dts-boxes

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

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

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

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

When we synchronize the same set of transitions in different orders, we get several resulting transitions with the same label and probability, but with different numberings having the same content. Then we shall consider only a single transition from the resulting ones in the plain dts-box to avoid introducing redundant ones.

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

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

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

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

For $E \in RegStatExpr$, let $Box_{dts}(\overline{E}) = \overline{Box_{dts}(E)}$ and $Box_{dts}(\underline{E}) = \underline{Box_{dts}(E)}$. This definition is compositional in the sense that, for any dynamic expression, we may decompose it in some inner dynamic and static expressions, for which we may apply the definition, thus obtaining the corresponding plain dtsi-boxes, which can be joined according to the term structure (by definition of Box_{dtsi}), the resulting plain box being marked in the places that were marked in the argument nets.

Let $\simeq$ denote the isomorphism between transition systems or between DTMCs and reachability graphs that binds their initial states. The names of transitions of the dts-box corresponding to a static expression could

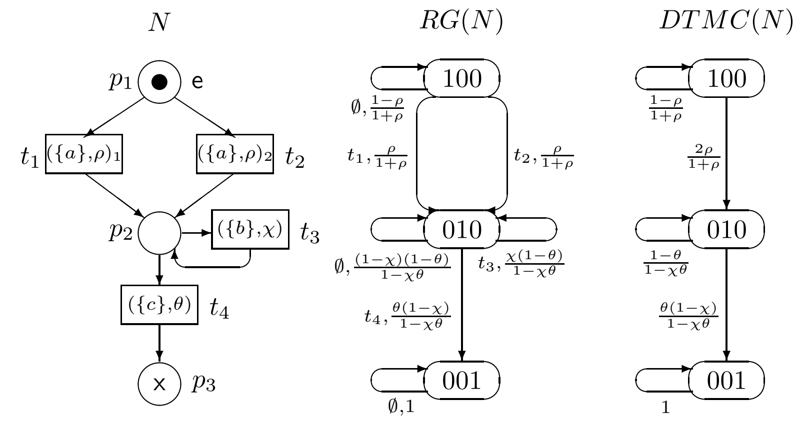

Figure 5: The marked dts-box $N = Box_{dts}(\overline{E})$ for $E = [((\{a\},\rho)_1[](\{a\},\rho)_2) * (\{b\},\chi) * (\{c\},\theta)]$, its reachability graph and the underlying DTMC

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 4.1 *[36] For any static expression E*

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

Proof. As for the qualitative behaviour, we have the same isomorphism as in PBC. The quantitative behaviour is the same, since the activities of an expression have probability parts coinciding with the probabilities of the transitions belonging to the corresponding dts-box and, both in stochastic processes specified by expressions and dts-boxes, conflicts are resolved via analogous probability functions. ☐

Proposition 4.1 *[36] For any static expression E*

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

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

Example 4.2 *Let E be from Example 3.2. 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 following example shows that without the syntactic restriction on regularity of expressions the corresponding marked dts-boxes may be not safe.

Example 4.3 *Let $E = [((\{a\},\frac{1}{2}) * ((\{b\},\frac{1}{2})\|(\{c\},\frac{1}{2})) * (\{d\},\frac{1}{2})]$. In Figure 6, the marked dts-box $N = Box_{dts}(\overline{E})$ and its reachability graph $RG(N)$ are presented. Symmetrically, in the marking $(0,1,1,2,0,0)$ there are 2 tokens in the place p_4. In the marking $(0,1,1,0,2,0)$ there are 2 tokens in the place p_5. Thus, allowing concurrency in the second argument of iteration in the expression $\overline{E}$ can lead to non-safeness of the corresponding marked dts-box N, though, it is 2-bounded in the worst case, see [4]. The origin of the problem is that N has as a self-loop with two subnets which can function independently. This explains why do we consider regular expressions only.*

5 Stochastic equivalences

Consider the expressions $E = (\{a\},\frac{1}{2})$ and $E' = (\{a\},\frac{1}{3})_1[](\{a\},\frac{1}{3})_2$, for which $\overline{E} \neq_{ts} \overline{E'}$, since $TS(\overline{E})$ has only one transition from the initial to the final state (with probability $\frac{1}{2}$) while $TS(\overline{E'})$ has two such ones (with probabilities $\frac{1}{4}$). On the other hand, all the mentioned transitions are labeled by activities with the same

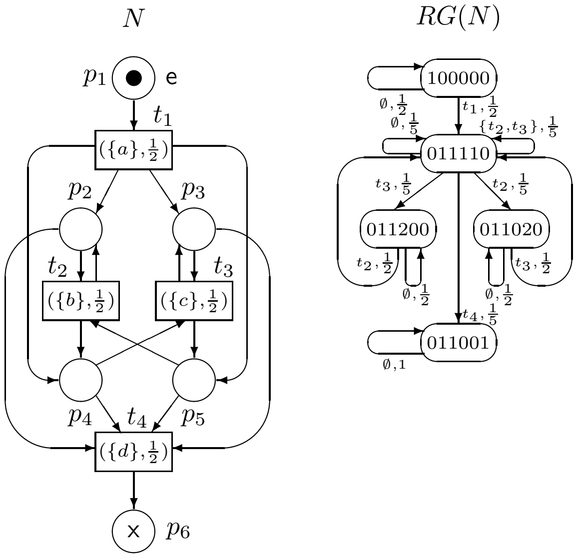

Figure 6: The marked dts-box $N = Box_{dtsi}(\overline{E})$ for $E = [((\{a\}, \frac{1}{2}) * ((\{b\}, \frac{1}{2})\|(\{c\}, \frac{1}{2})) * (\{d\}, \frac{1}{2})]$ and its reachability graph

multiaction part $\{a\}$. Moreover, the overall probabilities of the mentioned transitions of $TS(\overline{E})$ and $TS(\overline{E'})$ coincide: $\frac{1}{2} = \frac{1}{4} + \frac{1}{4}$. Further, $TS(\overline{E})$ (as well as $TS(\overline{E'})$) has one empty loop transition from the initial state to itself with probability $\frac{1}{2}$ and one empty loop transition from the final state to itself with probability 1. The empty loop transitions are labeled by the empty multiset of activities. Unlike $=_{ts}$, most of the probabilistic and stochastic equivalences proposed in the literature do not differentiate between the processes such as those specified by E and E'.

Since the semantic equivalence $=_{ts}$ is too discriminating in many cases, we need weaker equivalence notions. These equivalences should possess the following necessary properties. First, any two equivalent processes must have the same sequences of multisets of multiactions, which are the multiaction parts of the activities executed in steps starting from the initial states of the processes. Second, for every such sequence, its execution probabilities within both processes must coincide. Third, the desired equivalence should preserve the branching structure of computations, i.e., the points of choice of an external observer between several extensions of a particular computation should be taken into account. In this section, we define two such notions: step stochastic bisimulation equivalence and stochastic isomorphism.

5.1 Step stochastic bisimulation equivalence

Bisimulation equivalences respect the particular points of choice in the behavior of a system. To define stochastic bisimulation equivalences, we have to consider a bisimulation as an *equivalence* relation that partitions the states of the *union* of the transition systems $TS^*(G)$ and $TS^*(G')$ of two dynamic expressions G and G' to be compared. For G and G' to be bisimulation equivalent, the initial states of their transition systems, $[G]_\approx$ and $[G']_\approx$, are to be related by a bisimulation having the following transfer property: two states are related if in each of them the same multisets of multiactions can occur, and the resulting states *belong to the same equivalence class*. In addition, the sums of probabilities for all such occurrences should be the same for both states. Thus, we follow the approaches of [5, 17, 21].

In the definition below, we consider $\mathcal{L}(\Gamma) \in I\!N_f^{\mathcal{L}}$ for $\Gamma \in I\!N_f^{\mathcal{SL}}$, i.e., the (possibly empty) multisets of multiactions. The multiactions can be empty, then $\mathcal{L}(\Gamma)$ contains the elements $\emptyset$, and it is not empty itself.

Let G be a dynamic expression and $\mathcal{H} \subseteq DR(G)$. Then, for any $s \in DR(G)$ and $A \in I\!N_f^{\mathcal{L}}$, we write $s \xrightarrow{A}_{\mathcal{P}} \mathcal{H}$, where $\mathcal{P} = PM_A(s, \mathcal{H})$ is the *overall probability to move from s into the set of states $\mathcal{H}$ via steps with the multiaction part A* defined as

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

We write $s \xrightarrow{A} \mathcal{H}$ if $\exists \mathcal{P}\ s \xrightarrow{A}_{\mathcal{P}} \mathcal{H}$. Further, we write $s \rightarrow_{\mathcal{P}} \mathcal{H}$ if $\exists A\ s \xrightarrow{A} \mathcal{H}$, where $\mathcal{P} = PM(s, \mathcal{H})$ is the *overall probability to move from s into the set of states $\mathcal{H}$ via any steps* defined as

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

To introduce a stochastic bisimulation between dynamic expressions G and G', we should consider the "composite" set of states $DR(G)\cup DR(G')$, since we have to identify the probabilities to come from any two equivalent states into the same "composite" equivalence class (w.r.t. the stochastic bisimulation). Note that, for $G\neq G'$, transitions starting from the states of $DR(G)$ (or $DR(G')$) always lead to those from the same set, since $DR(G)\cap DR(G')=\emptyset$, and this allows us to "mix" the sets of states in the definition of stochastic bisimulation.

Definition 5.1 *Let G and G' be dynamic expressions. An* equivalence *relation $\mathcal{R}\subseteq(DR(G)\cup DR(G'))^2$ is a* step stochastic bisimulation *between G and G', denoted by $\mathcal{R}: G\underline{\leftrightarrow}_{ss}G'$, if:*

1. $([G]_\approx,[G']_\approx)\in\mathcal{R}$.
2. $(s_1,s_2)\in\mathcal{R}\ \Rightarrow\ \forall\mathcal{H}\in(DR(G)\cup DR(G'))/_{\mathcal{R}}\ \forall A\in I\!N_f^{\mathcal{L}}$

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

Dynamic expressions G and G' are step stochastic bisimulation equivalent*, denoted by $G\underline{\leftrightarrow}_{ss}G'$, if $\exists\mathcal{R}: G\underline{\leftrightarrow}_{ss}G'$.*

Let $\mathcal{R}_{ss}(G,G')=\bigcup\{\mathcal{R}\mid\mathcal{R}: G\underline{\leftrightarrow}_{ss}G'\}$ be the *union of all step stochastic bisimulations* between G and G'. The following proposition proves that $\mathcal{R}_{ss}(G,G')$ is also an *equivalence* and $\mathcal{R}_{ss}(G,G'): G\underline{\leftrightarrow}_{ss}G'$.

Proposition 5.1 *Let G and G' be dynamic expressions and $G\underline{\leftrightarrow}_{ss}G'$. Then $\mathcal{R}_{ss}(G,G')$ is the largest step stochastic bisimulation between G and G'.*

Proof. See Appendix A.1. □

5.2 Stochastic isomorphism

Stochastic isomorphism is weaker than $=_{ts}$. The main idea 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 $s\xrightarrow{A}_{\mathcal{P}}\{\tilde{s}\}$. Then we write $s\xrightarrow{A}_{\mathcal{P}}\tilde{s}$.

Definition 5.2 *Let G and G' be dynamic expressions. A mapping $\beta: DR(G)\to DR(G')$ is a* stochastic isomorphism *between G and G', denoted by $\beta: G=_{sto}G'$, if*

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

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

5.3 Interrelations of the stochastic equivalences

Now we compare the discrimination power of the stochastic equivalences.

Theorem 5.1 *For dynamic expressions G and G' the following strict implications hold:*

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

Proof. ($\Leftarrow$) Let us check the validity of the implications.

- The implication $=_{sto}\to\ \underline{\leftrightarrow}_{ss}$ is proved as follows. Let $\beta: G=_{sto}G'$. Then it is easy to see that $\mathcal{R}: G\underline{\leftrightarrow}_{ss}G'$, where $\mathcal{R}=\{(s,\beta(s))\mid s\in DR(G)\}$.
- The implication $=_{ts}\to=_{sto}$ is valid, since stochastic isomorphism is that of transition systems up to merging of transitions with labels having identical multiaction parts.
- The implication $\approx\to=_{ts}$ is valid, since the transition system of a dynamic formula is defined based on its structural equivalence class.

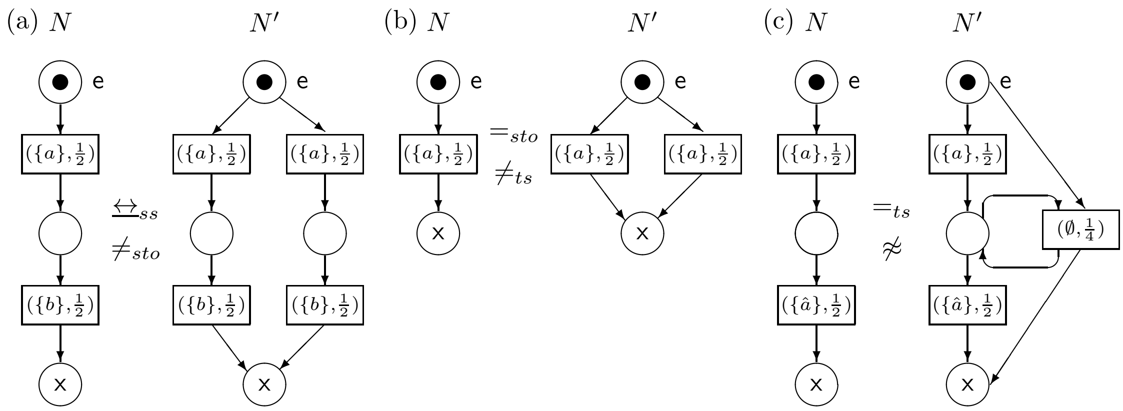

Figure 7: Dts-boxes of the dynamic expressions from equivalence examples of Theorem 5.1

($\Rightarrow$) The absence of additional nontrivial implications (not resulting from the combination of the existing ones) is proved by the following examples.

(a) Let $E = (\{a\}, \frac{1}{2}); (\{b\}, \frac{1}{2})$ and $E' = (\{a\}, \frac{1}{3}); (\{b\}, \frac{1}{2})[](\{a\}, \frac{1}{3}); (\{b\}, \frac{1}{2})$. Then $\overline{E} \underline{\leftrightarrow}_{ss} \overline{E'}$, but $\overline{E} \neq_{sto} \overline{E'}$, since $TS(\overline{E'})$ has more states than $TS(\overline{E})$.

(b) Let $E = (\{a\}, \frac{1}{2})$ and $E' = (\{a\}, \frac{1}{3})_1[](\{a\}, \frac{1}{3})_2$. Then $\overline{E} =_{sto} \overline{E'}$, but $\overline{E} \neq_{ts} \overline{E'}$, since $TS(\overline{E})$ has only one transition from the initial to the final state while $TS(\overline{E'})$ has two such ones.

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

□

Example 5.1 *In Figure 7, the marked dts-boxes corresponding to the dynamic expressions from equivalence examples of Theorem 5.1 are presented, i.e.,* $N = Box_{dts}(\overline{E})$ *and* $N' = Box_{dts}(\overline{E'})$ *for each picture (a)–(c).*

6 Reduction modulo equivalences

The equivalences which we proposed can be used to reduce transition systems and DTMCs of expressions (reachability graphs and DTMCs of dts-boxes). Reductions of graph-based models, like transition systems, reachability graphs and DTMCs, result in those with less states (the graph nodes). The goal of the reduction is to decrease the number of states in the semantic representation of the modeled system while preserving its important qualitative and quantitative properties. Thus, the reduction allows one to simplify the behaviour and performance analysis of systems.

An *autobisimulation* is a bisimulation between an expression and itself. For a dynamic expression G and a step stochastic autobisimulation on it $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G$, let $\mathcal{K} \in DR(G)/_{\mathcal{R}}$ and $s_1, s_2 \in \mathcal{K}$. We have $\forall \widetilde{\mathcal{K}} \in DR(G)/_{\mathcal{R}}\ \forall A \in I\!N_f^{\mathcal{L}}\ s_1 \xrightarrow{A}_{\mathcal{P}} \widetilde{\mathcal{K}} \Leftrightarrow s_2 \xrightarrow{A}_{\mathcal{P}} \widetilde{\mathcal{K}}$. The previous statement is valid for all $s_1, s_2 \in \mathcal{K}$, hence, we can rewrite it as $\mathcal{K} \xrightarrow{A}_{\mathcal{P}} \widetilde{\mathcal{K}}$, where $\mathcal{P} = PM_A(\mathcal{K}, \widetilde{\mathcal{K}}) = PM_A(s_1, \widetilde{\mathcal{K}}) = PM_A(s_2, \widetilde{\mathcal{K}})$.

We write $\mathcal{K} \xrightarrow{A} \widetilde{\mathcal{K}}$ if $\exists \mathcal{P}\ \mathcal{K} \xrightarrow{A}_{\mathcal{P}} \widetilde{\mathcal{K}}$ and $\mathcal{K} \to \widetilde{\mathcal{K}}$ if $\exists A\ \mathcal{K} \xrightarrow{A} \widetilde{\mathcal{K}}$. The similar arguments allow us to write $\mathcal{K} \to_{\mathcal{P}} \widetilde{\mathcal{K}}$, where $\mathcal{P} = PM(\mathcal{K}, \widetilde{\mathcal{K}}) = PM(s_1, \widetilde{\mathcal{K}}) = PM(s_2, \widetilde{\mathcal{K}})$.

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

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

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

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

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

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

Let $\mathcal{R}_{ss}(G) = \bigcup\{\mathcal{R} \mid \mathcal{R} : G\underline{\leftrightarrow}_{ss}G\}$ be the *union of all step stochastic autobisimulations* on G. By Proposition 5.1, $\mathcal{R}_{ss}(G)$ is the largest step stochastic autobisimulation on G. Based on the equivalence classes w.r.t. $\mathcal{R}_{ss}(G)$, the quotient (by $\underline{\leftrightarrow}_{ss}$) transition systems and the quotient (by $\underline{\leftrightarrow}_{ss}$) underlying DTMCs of expressions can be defined. The mentioned equivalence classes become the quotient states. Every quotient transition between two such composite states represents all steps (having the same multiaction part in case of the transition system quotient) from the first state to the second one.

Definition 6.1 *Let G be a dynamic expression. The* quotient (by $\underline{\leftrightarrow}_{ss}$) (labeled probabilistic) transition system *of G is a quadruple $TS_{\underline{\leftrightarrow}_{ss}}(G) = (S_{\underline{\leftrightarrow}_{ss}}, L_{\underline{\leftrightarrow}_{ss}}, \mathcal{T}_{\underline{\leftrightarrow}_{ss}}, s_{\underline{\leftrightarrow}_{ss}})$, where*

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

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

Definition 6.2 *Let G be a dynamic expression. The* quotient (by $\underline{\leftrightarrow}_{ss}$) underlying DTMC *of G, denoted by $DTMC_{\underline{\leftrightarrow}_{ss}}(G)$, has the state space $DR(G)/_{\mathcal{R}_{ss}(G)}$ and the transitions $\mathcal{K} \to_{\mathcal{P}} \widetilde{\mathcal{K}}$, where $\mathcal{P} = PM(\mathcal{K}, \widetilde{\mathcal{K}})$.*

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

The quotients of both transition systems and underlying DTMCs are the minimal reductions of the mentioned objects modulo $\underline{\leftrightarrow}_{ss}$. The quotients can be used to simplify analysis of system properties which are preserved by $\underline{\leftrightarrow}_{ss}$, since less states should be examined for it. Such reduction method resembles that from [1] based on place bisimulation equivalence for PNs. The algorithms which can be adapted for our framework exist for constructing the quotients of transition systems by bisimulation [34] and those of (discrete or continuous) Markov chains by ordinary lumping [13]. The algorithms have time complexity $O(m \lg n)$ and space complexity $O(m+n)$, where n is the number of states and m is the number of transitions. The comprehensive reduction examples will be presented in Section 8.

7 Stationary behaviour

Let us examine how the proposed equivalences can be used to compare the behaviour of stochastic processes in their steady states. We shall consider only formulas specifying stochastic processes with infinite behavior, i.e., expressions with the iteration operator. Note that the iteration operator does not guarantee infiniteness of behaviour, since there can exist a deadlock within the body (the second argument) of iteration when the corresponding subprocess does not reach its final state by some reasons.

Like in the framework of DTMCs, in DTSPNs the most common systems for performance analysis are *ergodic* (recurrent non-null, aperiodic and irreducible) ones. For ergodic DTSPNs, the steady-state marking probabilities exist and can be determined. In [24], the following sufficient (but not necessary) conditions for ergodicity of DTSPNs are stated: *liveness* (for each transition and any reachable marking there exist a sequence of markings from it leading to the marking enabling that transition), *boundedness* (the number of tokens in every place is not greater than some fixed number for any reachable marking) and *nondeterminism* (the transition probabilities are strictly less than 1). For the dts-box of a dynamic expression with no deadlocks in at least one of the bodies of the iteration operators it contains these three conditions are satisfied: the subnet corresponding to the deadlock-free iteration body is live, safe (1-bounded) and nondeterministic (since all markings of the live subnet are non-terminal, the probabilities of transitions from them are strictly less than 1). Hence, its DTMC restricted to the states between the initial and final states of the deadlock-free iteration body is ergodic. The isomorphism between DTMCs of expressions and those of the corresponding dts-boxes which is stated by Proposition 4.1 guarantees that the underlying DTMC of an expression with infinite behaviour is ergodic if restricted to the states in which a deadlock-free iteration body is executed.

In this section, we consider the expressions such that their underlined DTMCs contain one ergodic subset of states to guarantee that the single steady state exists.

7.1 Theoretical background

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

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

The transient (k-step, $k \in I\!N$) probability mass function (PMF) $\psi[k] = (\psi_1[k], \ldots, \psi_n[k])$ for $DTMC(G)$ is a solution of the equation system

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

where $\psi[0] = (\psi_1[0], \ldots, \psi_n[0])$ is the initial PMF defined as $\psi_i[0] = \begin{cases} 1, & s_i = [G]_{\approx}; \\ 0, & \text{otherwise.} \end{cases}$

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

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

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

where $\mathbf{E}$ is a unitary matrix of dimension n and $\mathbf{0}$ is a vector with n values 0, $\mathbf{1}$ is that with n values 1.

If $DTMC(G)$ has a single steady state then $\psi = \lim_{k\to\infty} \psi[k]$.

For $s = s_i \in DR(G)$ $(1 \leq i \leq n)$ let $\psi[k](s) = \psi_i[k]$ $(k \in I\!N)$ and $\psi(s) = \psi_i$.

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

- The *average recurrence (return) time in the state* s (i.e., the number of discrete time units or steps required for this) is $\frac{1}{\psi(s)}$.
- The *fraction of residence time in the state* s is $\psi(s)$.
- The *fraction of residence time in the set of states* $S \subseteq DR(G)$ or the *probability of the event determined by a condition that is true for all states from* S is $\sum_{s\in S} \psi(s)$.
- The *relative fraction of residence time in the set of states* S *w.r.t. that in* $\widetilde{S}$ is $\frac{\sum_{s\in S}\psi(s)}{\sum_{\tilde{s}\in\widetilde{S}}\psi(\tilde{s})}$.
- The *steady-state probability to perform a step with an activity* (α, ρ) is $\sum_{s\in DR(G)} \psi(s) \sum_{\{\Gamma|(\alpha,\rho)\in\Gamma\}} PT(\Gamma, s)$.
- The *probability of the event determined by a reward function* r is $\sum_{s\in DR(G)} \psi(s)r(s)$.

7.2 Steady state and equivalences

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

Proposition 7.1 *Let G, G' be dynamic expressions with $\mathcal{R} : G\underline{\leftrightarrow}_{ss}G'$. Then $\forall\mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$*

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

Proof. See Appendix A.2. □

By Proposition 7.1, $\underline{\leftrightarrow}_{ss}$ preserves the quantitative properties of the stationary behaviour. Now we intend to demonstrate that the qualitative properties of the stationary behaviour based on the multiaction labels are preserved as well.

Definition 7.1 *A* derived step trace *of a dynamic expression G is a chain $\Sigma = A_1 \cdots A_n \in (I\!N_f^{\mathcal{L}})^*$, where $\exists s \in DR(G)$ $s \stackrel{\Gamma_1}{\rightarrow} s_1 \stackrel{\Gamma_2}{\rightarrow} \cdots \stackrel{\Gamma_n}{\rightarrow} s_n$, $\mathcal{L}(\Gamma_i) = A_i$ $(1 \leq i \leq n)$. Then the* probability to execute the derived step trace Σ in s *is*

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

The following theorem demonstrates that, for two dynamic expressions related by $\underline{\leftrightarrow}_{ss}$, the steady-state probabilities to come in an equivalence class and start a derived step trace from it coincide.

Theorem 7.1 *Let G, G' be dynamic expressions with $\mathcal{R} : G\underline{\leftrightarrow}_{ss}G'$ and Σ be a derived step trace of G and G'. Then $\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$*

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

Proof. See Appendix A.3. □

Example 7.1 *The expression* $\mathsf{Stop} = (\{c\}, \frac{1}{2})$ rs c *specifies the non-terminating process that performs only empty loops with probability* 1. *Let* $E = [(\{a\}, \frac{1}{2}) * ((\{b\}, \frac{1}{2}); ((\{c\}, \frac{1}{3})_1[](\{c\}, \frac{1}{3})_2)) * \mathsf{Stop}]$ *and* $E' = [(\{a\}, \frac{1}{2}) * (((\{b\}, \frac{1}{3})_1; (\{c\}, \frac{1}{2})_1)[]((\{b\}, \frac{1}{3})_2; (\{c\}, \frac{1}{2})_2)) * \mathsf{Stop}]$. *We have* $\overline{E} =_{sto} \overline{E'}$, *hence,* $\overline{E}\underline{\leftrightarrow}_{ss}\overline{E'}$.

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

$$s_1 = [[\overline{(\{a\}, \tfrac{1}{2})} * ((\{b\}, \tfrac{1}{2}); ((\{c\}, \tfrac{1}{3})_1[](\{c\}, \tfrac{1}{3})_2)) * \mathsf{Stop}]]_{\approx},$$
$$s_2 = [[(\{a\}, \tfrac{1}{2}) * (\overline{(\{b\}, \tfrac{1}{2})}; ((\{c\}, \tfrac{1}{3})_1[](\{c\}, \tfrac{1}{3})_2)) * \mathsf{Stop}]]_{\approx},$$
$$s_3 = [[(\{a\}, \tfrac{1}{2}) * ((\{b\}, \tfrac{1}{2}); \overline{((\{c\}, \tfrac{1}{3})_1[](\{c\}, \tfrac{1}{3})_2))} * \mathsf{Stop}]]_{\approx}.$$

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

$$s_1' = [[\overline{(\{a\}, \tfrac{1}{2})} * (((\{b\}, \tfrac{1}{3})_1; (\{c\}, \tfrac{1}{2})_1)[]((\{b\}, \tfrac{1}{3})_2; (\{c\}, \tfrac{1}{2})_2)) * \mathsf{Stop}]]_{\approx},$$
$$s_2' = [[(\{a\}, \tfrac{1}{2}) * \overline{(((\{b\}, \tfrac{1}{3})_1; (\{c\}, \tfrac{1}{2})_1)[]((\{b\}, \tfrac{1}{3})_2; (\{c\}, \tfrac{1}{2})_2))} * \mathsf{Stop}]]_{\approx},$$
$$s_3' = [[(\{a\}, \tfrac{1}{2}) * (((\{b\}, \tfrac{1}{3})_1; \overline{(\{c\}, \tfrac{1}{2})_1})[]((\{b\}, \tfrac{1}{3})_2; (\{c\}, \tfrac{1}{2})_2)) * \mathsf{Stop}]]_{\approx},$$
$$s_4' = [[(\{a\}, \tfrac{1}{2}) * (((\{b\}, \tfrac{1}{3})_1; (\{c\}, \tfrac{1}{2})_1)[]((\{b\}, \tfrac{1}{3})_2; \overline{(\{c\}, \tfrac{1}{2})_2})) * \mathsf{Stop}]]_{\approx}.$$

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

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

Consider the equivalence class (w.r.t. $\mathcal{R}_{ss}(\overline{E}, \overline{E'})$) $\mathcal{H} = \{s_3, s_3', s_4'\}$. One can see that the steady-state probabilities for $\mathcal{H}$ coincide: $\sum_{s\in\mathcal{H}\cap DR(\overline{E})} \psi(s) = \psi(s_3) = \frac{1}{2} = \frac{1}{4} + \frac{1}{4} = \psi'(s_3') + \psi'(s_4') = \sum_{s'\in\mathcal{H}\cap DR(\overline{E'})} \psi'(s')$. Let $\Sigma = \{\{c\}\}$. The steady-state probabilities to come in the equivalence class $\mathcal{H}$ and start the derived step trace Σ from it coincide as well: $\psi(s_3)(PT(\{(\{c\}, \frac{1}{3})_1\}, s_3) + PT(\{(\{c\}, \frac{1}{3})_2\}, s_3)) = \frac{1}{2}\left(\frac{1}{4} + \frac{1}{4}\right) = \frac{1}{4} = \frac{1}{4} \cdot \frac{1}{2} + \frac{1}{4} \cdot \frac{1}{2} = \psi'(s_3')PT(\{(\{c\}, \frac{1}{2})_1\}, s_3') + \psi'(s_4')PT(\{(\{c\}, \frac{1}{2})_2\}, s_4')$.

In Figure 8, the marked dts-boxes corresponding to the dynamic expressions above are presented, i.e., $N = Box_{dts}(\overline{E})$ and $N' = Box_{dts}(\overline{E'})$.

7.3 Preservation of performance and simplification of its analysis

Many performance indices are based on the steady-state probabilities to come in a set of similar states or, after coming in it, to start a step trace from this set. The similarity of states is usually captured by an equivalence relation, hence, the sets are often the equivalence classes. Proposition 7.1 and Theorem 7.1 guarantee a coincidence of the mentioned indices for the expressions related by $\underline{\leftrightarrow}_{ss}$. Thus, $\underline{\leftrightarrow}_{ss}$ (hence, all the stronger equivalences we have considered) preserves performance of stochastic systems modeled by expressions of dtsPBC.

In addition, it is easier to evaluate performance using a DTMC with less states, since in this case the dimension of the transition probability matrix is smaller, and we solve systems of less equations to calculate steady-state probabilities. The reasoning above validates the following method of performance analysis simplification.

1. The investigated system is specified by a static expression of dtsPBC.

2. The transition system of the expression is constructed.

3. After treating the transition system for self-similarity, a step stochastic autobisimulation equivalence for the expression is determined.

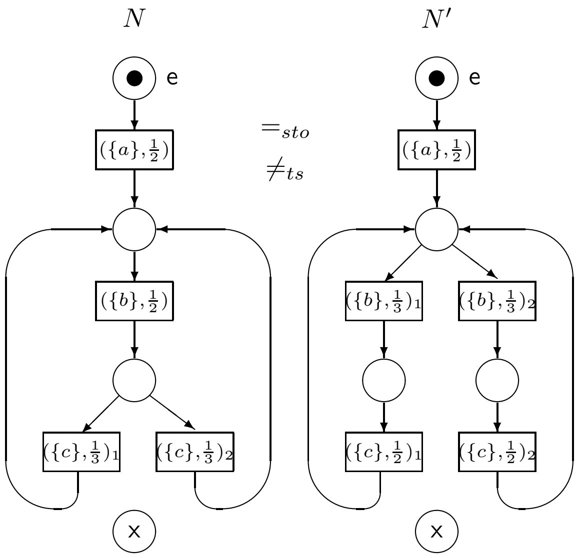

Figure 8: $\underline{\leftrightarrow}_{ss}$ implies a coincidence of the steady-state probabilities to come in an equivalence class and start a derived trace from it

4. The quotient underlying DTMC is constructed.

5. Stationary probabilities and performance indices are calculated via the DTMC.

The limitation of the method above is its applicability only to the expressions such that their corresponding DTMCs contain one irreducible subset of states, i.e., the existence of exactly one stationary state is required. If a DTMC contains several irreducible subsets of states then several steady states can exist which depend on the initial PMF. There is an analytical method to determine the stable states for DTMCs of this kind as well. Note that, for every expression, the underlying DTMC has by definition only one initial PMF (that at the time moment 0), hence, the stationary state will be only one in this case too. In addition, it is worth to apply the method only to the systems with similar subprocesses.

8 Dining philosophers system

8.1 The standard system

Consider a model of five dining philosophers, for which the Petri net interpretation was proposed in [32]. We investigate this dining philosophers system in the discrete time stochastic setting of dtsPBC. The philosophers occupy a round table, and there is one fork between every neighboring persons, hence, there are five forks on the table. A philosopher needs two forks to eat, namely, his left and right ones. Hence, all five philosophers cannot eat together, since otherwise there will not be enough forks available, but only one of two of them who are not neighbors. The model performs as follows. After the activation of the system (the philosophers come in the dining room), five forks are placed on the table. If the left and right forks are available for a philosopher, he takes them simultaneously and begins eating. At the end of eating, the philosopher places both his forks simultaneously back on the table. The strategy to pick up and release two forks simultaneously prevents the situation when a philosopher takes one fork but is not able to pick up the second one since their neighbor has already done so. In particular, we avoid a deadlock when all the philosophers take their left (right) forks and wait until their right (left) forks will be available. Figure 9 presents the diagram of the system.

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

Since the neighbors cannot dine together, the maximal number of the dining persons for the system with n philosophers will be $\lfloor \frac{n}{2} \rfloor$, i.e., the maximal natural number that is not greater than $\frac{n}{2}$. Note that if the philosopher i belongs to some maximal set then the philosopher $i(\mathsf{mod}\ n)+1$ will belong to the next one. Let

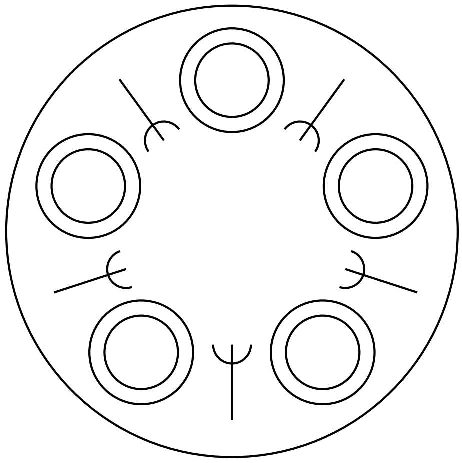

Figure 9: The diagram of the dining philosophers system

us calculate how many such different maximal sets are there. If n is an even number then there will be only 2 maximal sets of $\frac{n}{2}$ dining persons, namely, the philosophers numbered with all odd natural numbers which are not greater than n and those numbered with all even natural numbers which are not greater than n. If n is an odd number then there will be n maximal sets of $\frac{n-1}{2}$ dining persons, since, starting from some maximal set one can "shift" clockwise $n-1$ times by one element modulo n until the next maximal set will coincide with the initial one.

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

The static expression of the philosopher i ($1 \leq i \leq 4$) is
$E_i = [(\{x_i\}, \frac{1}{2}) * (((\{b_i, \widehat{y_i}\}, \frac{1}{2}); (\{e_i, \widehat{z_i}\}, \frac{1}{2}))[]((\{y_{i+1}\}, \frac{1}{2}); (\{z_{i+1}\}, \frac{1}{2}))) * \mathsf{Stop}]$.

The static expression of the philosopher 5 is
$E_5 = [(\{a, \widehat{x_1}, \widehat{x_2}, \widehat{x_2}, \widehat{x_4}\}, \frac{1}{2}) * (((\{b_5, \widehat{y_5}\}, \frac{1}{2}); (\{e_5, \widehat{z_5}\}, \frac{1}{2}))[]((\{y_1\}, \frac{1}{2}); (\{z_1\}, \frac{1}{2}))) * \mathsf{Stop}]$.

The static expression of the dining philosophers system is
$E = (E_1 \| E_2 \| E_3 \| E_4 \| E_5)$ $\mathsf{sy}\ x_1$ $\mathsf{sy}\ x_2$ $\mathsf{sy}\ x_3$ $\mathsf{sy}\ x_4$ $\mathsf{sy}\ y_1$ $\mathsf{sy}\ y_2$ $\mathsf{sy}\ y_3$ $\mathsf{sy}\ y_4$ $\mathsf{sy}\ y_5$ $\mathsf{sy}\ z_1$ $\mathsf{sy}\ z_2$ $\mathsf{sy}\ z_3$ $\mathsf{sy}\ z_4$ $\mathsf{sy}\ z_5$ $\mathsf{rs}\ x_1$ $\mathsf{rs}\ x_2$ $\mathsf{rs}\ x_3$ $\mathsf{rs}\ x_4$ $\mathsf{rs}\ y_1$ $\mathsf{rs}\ y_2$ $\mathsf{rs}\ y_3$ $\mathsf{rs}\ y_4$ $\mathsf{rs}\ y_5$ $\mathsf{rs}\ z_1$ $\mathsf{rs}\ z_2$ $\mathsf{rs}\ z_3$ $\mathsf{rs}\ z_4$ $\mathsf{rs}\ z_5$.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$DR(\overline{E})$ has 12 states interpreted as follows: s_1 is the initial state, s_2: the system is activated and no philosophers dine, s_3: philosopher 1 dines, s_4: philosophers 1 and 4 dine, s_5: philosophers 1 and 3 dine, s_6: philosopher 4 dines, s_7: philosopher 3 dines, s_8: philosophers 2 and 4 dine, s_9: philosophers 3 and 5 dine, s_{10}: philosopher 2 dines, s_{11}: philosopher 5 dines, s_{12}: philosophers 2 and 5 dine.

In Figure 10, the transition system $TS(\overline{E})$ is presented.

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

$$SJ = \left(32, \frac{29}{20}, \frac{20}{11}, \frac{16}{7}, \frac{16}{7}, \frac{20}{11}, \frac{20}{11}, \frac{16}{7}, \frac{16}{7}, \frac{20}{11}, \frac{20}{11}, \frac{16}{7}\right).$$

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

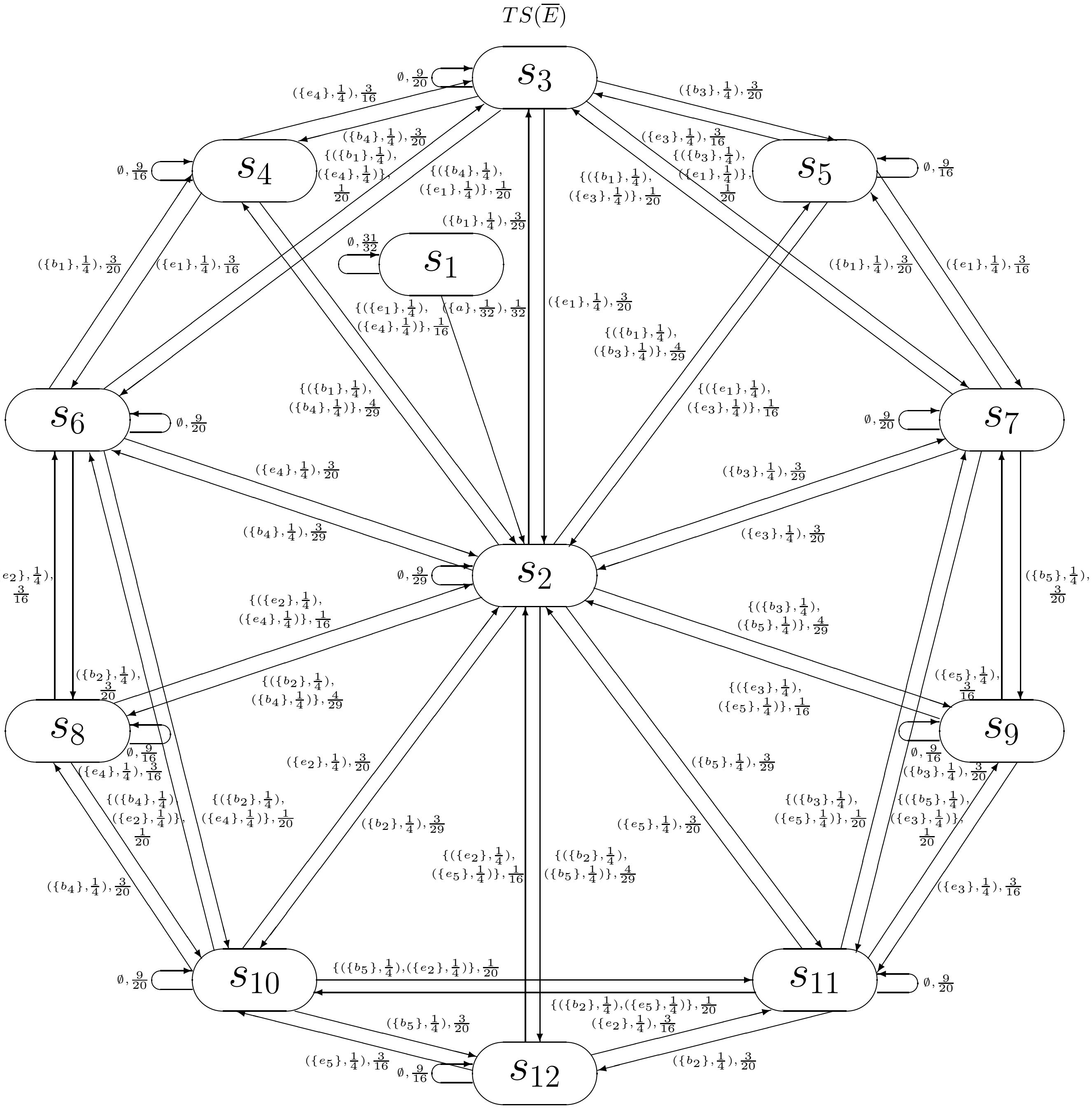

Figure 10: The transition system of the dining philosophers system

Table 4: Transient and steady-state probabilities of the dining philosophers system

k	0	20	40	60	80	100	120	140	160	180	200	∞
$\psi_1[k]$	1	0.5299	0.2808	0.1488	0.0789	0.0418	0.0222	0.0117	0.0062	0.0033	0.0017	0
$\psi_2[k]$	0	0.0842	0.1098	0.1234	0.1306	0.1345	0.1365	0.1375	0.1381	0.1384	0.1386	0.1388
$\psi_3[k]$	0	0.0437	0.0681	0.0811	0.0880	0.0916	0.0935	0.0945	0.0951	0.0954	0.0955	0.0957
$\psi_4[k]$	0	0.0335	0.0537	0.0645	0.0701	0.0732	0.0748	0.0756	0.0760	0.0763	0.0764	0.0766

$$VAR = \left(1024, \frac{841}{400}, \frac{400}{121}, \frac{256}{49}, \frac{256}{49}, \frac{400}{121}, \frac{400}{121}, \frac{256}{49}, \frac{256}{49}, \frac{400}{121}, \frac{400}{121}, \frac{256}{49}\right).$$

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

$$\mathbf{P} = \begin{bmatrix}
\frac{31}{32} & \frac{1}{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{9}{29} & \frac{3}{29} & \frac{1}{29} & \frac{1}{29} & \frac{3}{29} & \frac{3}{29} & \frac{1}{29} & \frac{1}{29} & \frac{3}{29} & \frac{3}{29} & \frac{1}{29} \\
0 & \frac{3}{20} & \frac{9}{20} & \frac{3}{20} & \frac{3}{20} & \frac{1}{20} & \frac{1}{20} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{16} & \frac{3}{16} & \frac{9}{16} & 0 & \frac{3}{16} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{16} & \frac{3}{16} & 0 & \frac{9}{16} & 0 & \frac{3}{16} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{3}{20} & \frac{1}{20} & \frac{3}{20} & 0 & \frac{9}{20} & 0 & \frac{3}{20} & 0 & \frac{1}{20} & 0 & 0 \\
0 & \frac{3}{20} & \frac{1}{20} & 0 & \frac{3}{20} & 0 & \frac{9}{20} & 0 & \frac{3}{20} & 0 & \frac{1}{20} & 0 \\
0 & \frac{1}{16} & 0 & 0 & 0 & \frac{3}{16} & 0 & \frac{9}{16} & 0 & \frac{3}{16} & 0 & 0 \\
0 & \frac{1}{16} & 0 & 0 & 0 & 0 & \frac{3}{16} & 0 & \frac{9}{16} & 0 & \frac{3}{16} & 0 \\
0 & \frac{3}{20} & 0 & 0 & 0 & \frac{1}{20} & 0 & \frac{3}{20} & 0 & \frac{9}{20} & \frac{1}{20} & \frac{3}{20} \\
0 & \frac{3}{20} & 0 & 0 & 0 & 0 & \frac{1}{20} & 0 & \frac{3}{20} & \frac{1}{20} & \frac{9}{20} & \frac{3}{20} \\
0 & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{16} & \frac{3}{16} & \frac{9}{16}
\end{bmatrix}.$$

In Table 4, the transient and the steady-state probabilities $\psi_i[k]$ $(1 \le i \le 4)$ of the dining philosophers system at the time moments $k \in \{0, 20, 40, \ldots, 200\}$ and $k = \infty$ are presented, and in Figure 11, the alteration diagram (evolution in time) for the transient probabilities is depicted. It is sufficient to consider the probabilities for the states $s_1, \ldots, s_4$ only, since the corresponding values coincide for $s_3, s_6, s_7, s_{10}, s_{11}$, as well as for $s_4, s_5, s_8, s_9, s_{12}$.

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

$$\psi = \left(0, \frac{29}{209}, \frac{20}{209}, \frac{16}{209}, \frac{16}{209}, \frac{20}{209}, \frac{20}{209}, \frac{16}{209}, \frac{16}{209}, \frac{20}{209}, \frac{20}{209}, \frac{16}{209}\right).$$

We can now calculate the main performance indices.

- The average recurrence time in the state s_2, where all the forks are available, called the *average system run-through*, is $\frac{1}{\psi_2} = \frac{209}{29} = 7\frac{6}{29}$.
- Nobody eats in the state s_2. Then, the *fraction of time when no philosophers dine* is $\psi_2 = \frac{29}{209}$.

  Only one philosopher eats in the states $s_3, s_6, s_7, s_{10}, s_{11}$. Then, the *fraction of time when only one philosopher dines* is $\psi_3 + \psi_6 + \psi_7 + \psi_{10} + \psi_{11} = \frac{20}{209} + \frac{20}{209} + \frac{20}{209} + \frac{20}{209} + \frac{20}{209} = \frac{100}{209}$.

  Two philosophers eat together in the states $s_4, s_5, s_8, s_9, s_{12}$. Then, the *fraction of time when two philosophers dine* is $\psi_4 + \psi_5 + \psi_8 + \psi_9 + \psi_{12} = \frac{16}{209} + \frac{16}{209} + \frac{16}{209} + \frac{16}{209} + \frac{16}{209} = \frac{80}{209}$.

  The *relative fraction of time when two philosophers dine w.r.t. when only one philosopher dines* is $\frac{80}{209} \cdot \frac{209}{100} = \frac{4}{5}$.
- The beginning of eating of first philosopher $(\{b_1\}, \frac{1}{4})$ is only possible from the states s_2, s_6, s_7. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of activities containing $(\{b_1\}, \frac{1}{4})$. Thus, the *steady-state probability of the beginning of eating of first philosopher* is $\psi_2 \sum_{\{\Gamma \mid (\{b_1\}, \frac{1}{4}) \in \Gamma\}} PT(\Gamma, s_2) + \psi_6 \sum_{\{\Gamma \mid (\{b_1\}, \frac{1}{4}) \in \Gamma\}} PT(\Gamma, s_6) + \psi_7 \sum_{\{\Gamma \mid (\{b_1\}, \frac{1}{4}) \in \Gamma\}} PT(\Gamma, s_7) = \frac{29}{209}\left(\frac{3}{29} + \frac{1}{29} + \frac{1}{29}\right) + \frac{20}{209}\left(\frac{3}{20} + \frac{1}{20}\right) + \frac{20}{209}\left(\frac{3}{20} + \frac{1}{20}\right) = \frac{13}{209}$.

In Figure 12, the marked dts-boxes corresponding to the dynamic expressions of the dining philosophers are presented, i.e., $N_i = Box_{dts}(\overline{E_i})$ $(1 \le i \le 5)$. In Figure 13, the marked dts-box corresponding to the dynamic expression of the dining philosophers system is depicted, i.e., $N = Box_{dts}(\overline{E})$.

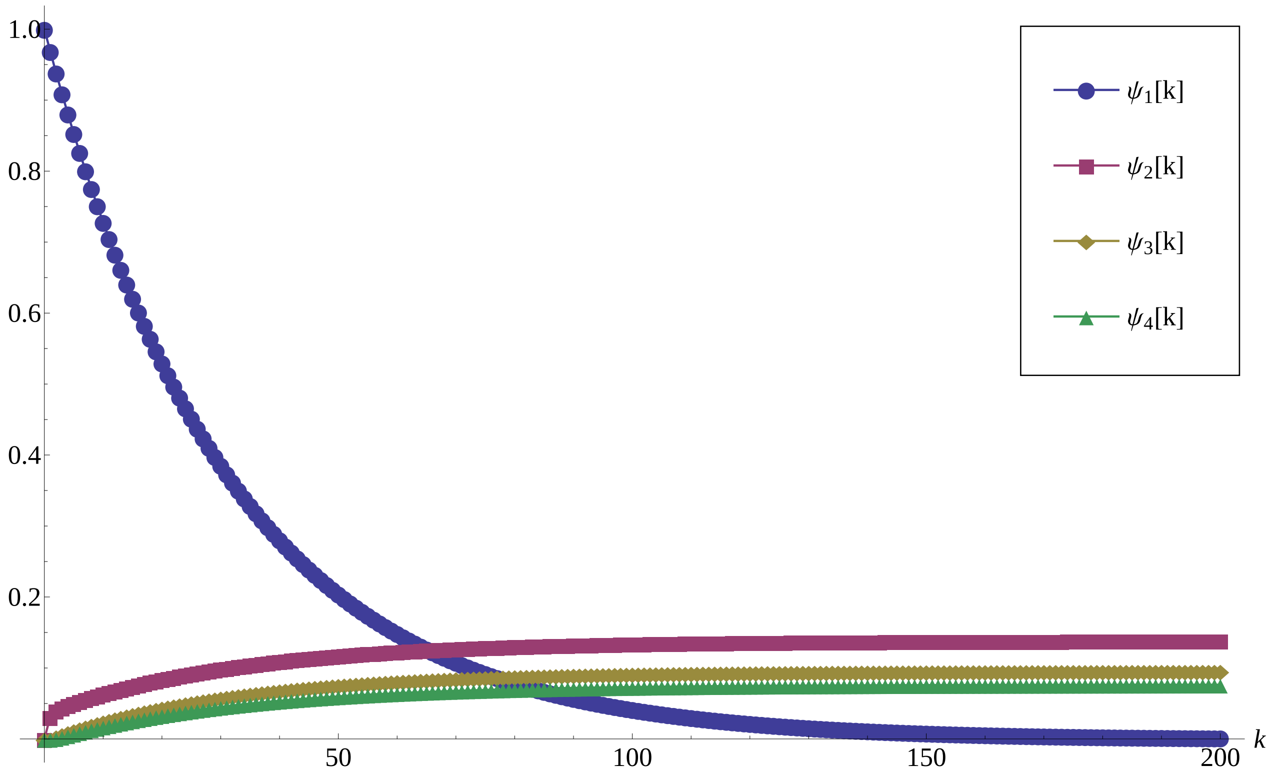

Figure 11: Transient probabilities alteration diagram of the dining philosophers system

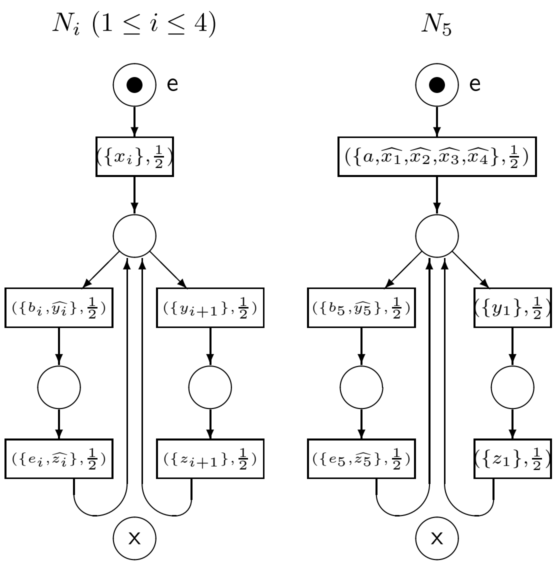

Figure 12: The marked dts-boxes of the dining philosophers

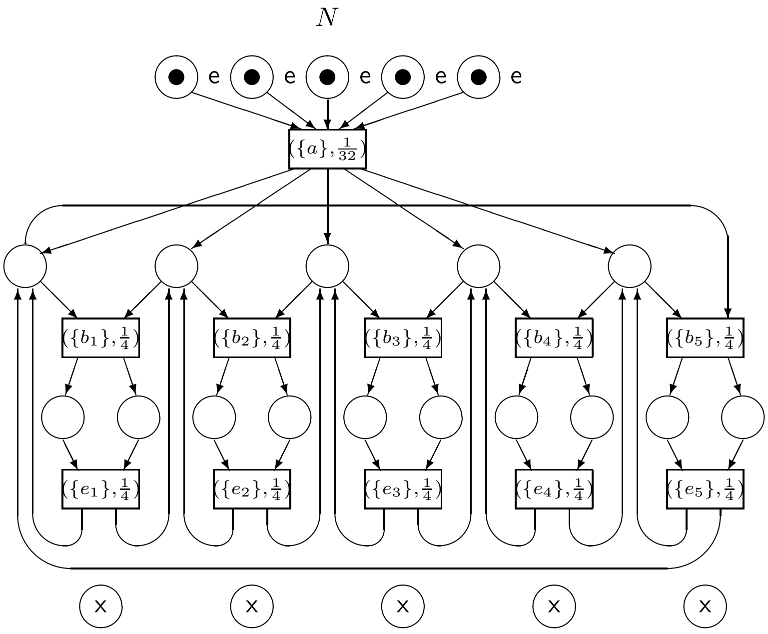

Figure 13: The marked dts-box of the dining philosophers system

8.2 The abstract system and its reduction

Let us consider a modification of the dining philosophers system with abstraction from personalities such that all the philosophers are indistinguishable. For example, we can just see that one or two philosophers dine but cannot observe who they are. We call this system the abstract dining philosophers one.

The static expression of the philosopher i $(1 \le i \le 4)$ is
$F_i = [(\{x_i\}, \frac{1}{2}) * (((\{b, \widehat{y_i}\}, \frac{1}{2}); (\{e, \widehat{z_i}\}, \frac{1}{2}))[]((\{y_{i+1}\}, \frac{1}{2}); (\{z_{i+1}\}, \frac{1}{2}))) * \mathsf{Stop}]$.

The static expression of the philosopher 5 is
$F_5 = [(\{a, \widehat{x_1}, \widehat{x_2}, \widehat{x_2}, \widehat{x_4}\}, \frac{1}{2}) * (((\{b, \widehat{y_5}\}, \frac{1}{2}); (\{e, \widehat{z_5}\}, \frac{1}{2}))[]((\{y_1\}, \frac{1}{2}); (\{z_1\}, \frac{1}{2}))) * \mathsf{Stop}]$.

The static expression of the abstract dining philosophers system is
$F = (F_1 \| F_2 \| F_3 \| F_4 \| F_5)$ sy x_1 sy x_2 sy x_3 sy x_4 sy y_1 sy y_2 sy y_3 sy y_4 sy y_5 sy z_1 sy z_2 sy z_3 sy z_4 sy z_5 rs x_1 rs x_2 rs x_3 rs x_4 rs y_1 rs y_2 rs y_3 rs y_4 rs y_5 rs z_1 rs z_2 rs z_3 rs z_4 rs z_5.

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

The first performance index and the second group of them coincide for the standard and the abstract systems. The following performance index based on non-personalized viewpoint to the philosophers.

- The beginning of eating of a philosopher $(\{b\}, \frac{1}{4})$ is only possible from the states $s_2, s_3, s_6, s_7, s_{10}, s_{11}$. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of activities containing $(\{b\}, \frac{1}{4})$. Thus, the *steady-state probability of the beginning of eating of a philosopher* is $\psi_2 \sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}} PT(\Gamma, s_2) + \psi_3 \sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}} PT(\Gamma, s_3) + \psi_6 \sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}} PT(\Gamma, s_6) + \psi_7 \sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}} PT(\Gamma, s_7) + \psi_{10} \sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}} PT(\Gamma, s_{10}) + \psi_{11} \sum_{\{\Gamma|(\{b\},\frac{1}{4})\in\Gamma\}} PT(\Gamma, s_{11}) = \frac{29}{209}\left(\frac{3}{29} + \frac{1}{29} + \frac{3}{29} + \frac{1}{29} + \frac{3}{29} + \frac{1}{29} + \frac{3}{29} + \frac{1}{29} + \frac{3}{29} + \frac{1}{29}\right) + \frac{20}{209}\left(\frac{3}{20} + \frac{1}{20} + \frac{3}{20} + \frac{1}{20}\right) + \frac{20}{209}\left(\frac{3}{20} + \frac{1}{20} + \frac{3}{20} + \frac{1}{20}\right) + \frac{20}{209}\left(\frac{3}{20} + \frac{1}{20} + \frac{3}{20} + \frac{1}{20}\right) + \frac{20}{209}\left(\frac{3}{20} + \frac{1}{20} + \frac{3}{20} + \frac{1}{20}\right) + \frac{20}{209}\left(\frac{3}{20} + \frac{1}{20} + \frac{3}{20} + \frac{1}{20}\right) = \frac{60}{209}$.

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

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

In Figure 14, the quotient transition system $TS_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is presented.

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

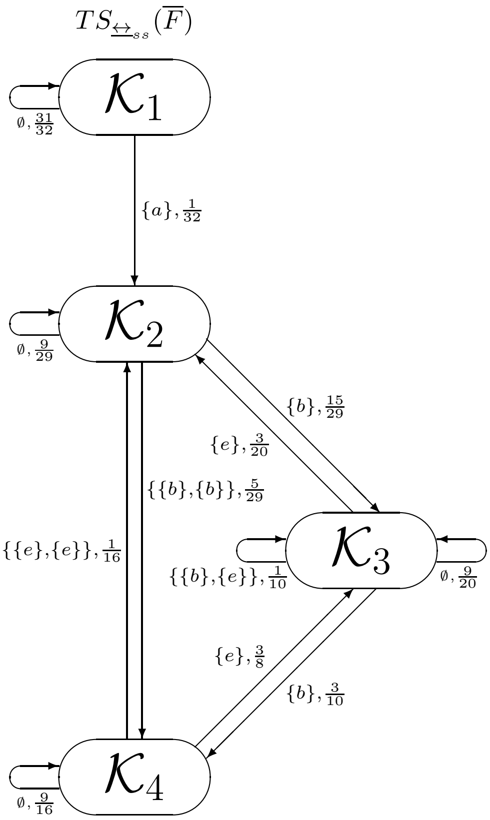

Figure 14: The quotient transition system of the abstract dining philosophers system

Table 5: Transient and steady-state probabilities of the quotient abstract dining philosophers system

k	0	20	40	60	80	100	120	140	160	180	200	∞
$\psi'_1[k]$	1	0.5299	0.2808	0.1488	0.0789	0.0418	0.0222	0.0117	0.0062	0.0033	0.0017	0
$\psi'_2[k]$	0	0.0842	0.1098	0.1234	0.1306	0.1345	0.1365	0.1375	0.1381	0.1384	0.1386	0.1388
$\psi'_3[k]$	0	0.2183	0.3406	0.4054	0.4398	0.4580	0.4676	0.4727	0.4754	0.4769	0.4776	0.4785
$\psi'_4[k]$	0	0.1675	0.2687	0.3223	0.3507	0.3658	0.3738	0.3780	0.3802	0.3814	0.3821	0.3828

$$SJ = \left(32, \frac{29}{20}, \frac{20}{9}, \frac{16}{7}\right).$$

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

$$VAR = \left(1024, \frac{841}{400}, \frac{400}{81}, \frac{256}{49}\right).$$

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

$$\mathbf{P}' = \begin{bmatrix} \frac{31}{32} & \frac{1}{32} & 0 & 0 \\ 0 & \frac{9}{29} & \frac{15}{29} & \frac{5}{29} \\ 0 & \frac{3}{20} & \frac{11}{20} & \frac{3}{10} \\ 0 & \frac{1}{16} & \frac{3}{8} & \frac{9}{16} \end{bmatrix}.$$

In Table 5, the transient and the steady-state probabilities $\psi'_i[k]$ $(1 \leq i \leq 4)$ of the quotient abstract dining philosophers system at the time moments $k \in \{0, 20, 40, \ldots, 200\}$ and $k = \infty$ are presented, and in Figure 15, the alteration diagram (evolution in time) for the transient probabilities is depicted.

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

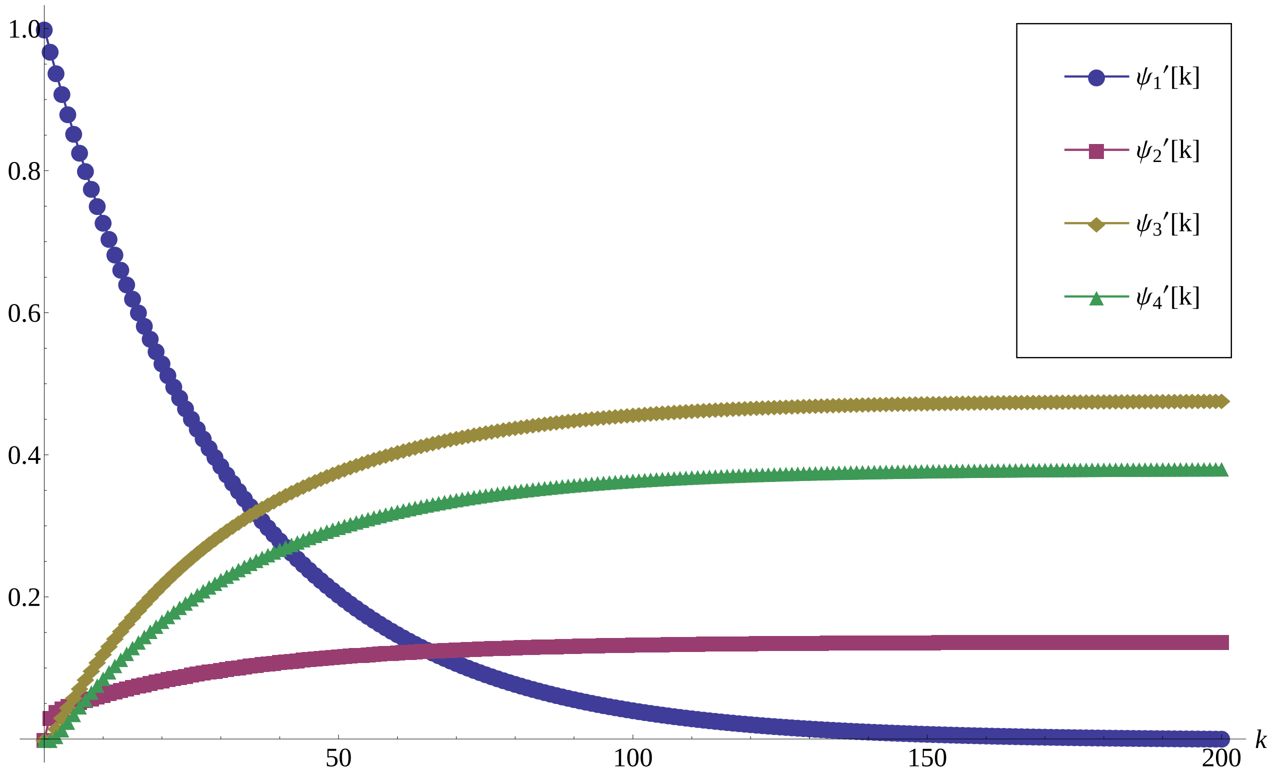

Figure 15: Transient probabilities alteration diagram of the quotient abstract dining philosophers system

$$\psi' = \left(0, \frac{29}{209}, \frac{100}{209}, \frac{80}{209}\right).$$

We can now calculate the main performance indices.

- The average recurrence time in the state $\mathcal{K}_2$, where all the forks are available, called the *average system run-through*, is $\frac{1}{\psi'_2} = \frac{209}{29} = 7\frac{6}{29}$.
- Nobody eats in the state $\mathcal{K}_2$. The *fraction of time when no philosophers dine* is $\psi'_2 = \frac{29}{209}$.

  Only one philosopher eats in the state $\mathcal{K}_3$. The *fraction of time when only one philosopher dines* is $\psi'_3 = \frac{100}{209}$.

  Two philosophers eat together in the state $\mathcal{K}_4$. The *fraction of time when two philosophers dine* is $\psi'_4 = \frac{80}{209}$.

  The *relative fraction of time when two philosophers dine w.r.t. when only one philosopher dines* is $\frac{80}{209} \cdot \frac{209}{100} = \frac{4}{5}$.
- The beginning of eating of a philosopher $\{b\}$ is only possible from the states $\mathcal{K}_2, \mathcal{K}_3$. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of multiactions containing $\{b\}$. Thus, the *steady-state probability of the beginning of eating of a philosopher* is $\psi'_2 \sum_{\{A,\widetilde{\mathcal{K}} \mid \{b\}\in A,\ \mathcal{K}_2 \xrightarrow{A} \widetilde{\mathcal{K}}\}} PM_A(\mathcal{K}_2, \widetilde{\mathcal{K}}) + \psi'_3 \sum_{\{A,\widetilde{\mathcal{K}} \mid \{b\}\in A,\ \mathcal{K}_3 \xrightarrow{A} \widetilde{\mathcal{K}}\}} PM_A(\mathcal{K}_3, \widetilde{\mathcal{K}}) = \frac{29}{209}\left(\frac{15}{29} + \frac{5}{29}\right) + \frac{100}{209}\left(\frac{3}{10} + \frac{1}{10}\right) = \frac{60}{209}$.

Observe that the performance indices are the same for the complete and the quotient abstract dining philosophers systems. The coincidence of the first performance index, as well as the second group of indices obviously illustrates the result of Proposition 7.1. The coincidence of the third performance index is due to Theorem 7.1: one should just apply its result to the step traces $\{\{b\}\}$, $\{\{b\},\{b\}\}$, $\{\{b\},\{e\}\}$ of the expressions $\overline{F}$ and $\overline{F'}$, and then sum the left and right parts of the three resulting equalities.

8.3 The generalized system

Let us determine which is the influence of the multiaction probabilities from specification of the dining philosophers system on its performance. Let all these multiactions have the same probability ρ. The resulting specification K of the generalized dining philosophers system is defined as follows.

The static expression of the philosopher i $(1 \leq i \leq 4)$ is
$K_i = [(\{x_i\}, \rho) * (((\{b_i, \widehat{y_i}\}, \rho); (\{e_i, \widehat{z_i}\}, \rho))[]((\{y_{i+1}\}, \rho); (\{z_{i+1}\}, \rho))) * \mathsf{Stop}]$.

The static expression of the philosopher 5 is
$K_5 = [(\{a, \widehat{x_1}, \widehat{x_2}, \widehat{x_2}, \widehat{x_4}\}, \rho) * (((\{b_5, \widehat{y_5}\}, \rho); (\{e_5, \widehat{z_5}\}, \rho))[]((\{y_1\}, \rho); (\{z_1\}, \rho))) * \mathsf{Stop}]$.

The static expression of the generalized dining philosophers system is
$K = (K_1 \| K_2 \| K_3 \| K_4 \| K_5)$ sy x_1 sy x_2 sy x_3 sy x_4 sy y_1 sy y_2 sy y_3 sy y_4 sy y_5 sy z_1 sy z_2 sy z_3 sy z_4 sy z_5 rs x_1 rs x_2 rs x_3 rs x_4 rs y_1 rs y_2 rs y_3 rs y_4 rs y_5 rs z_1 rs z_2 rs z_3 rs z_4 rs z_5.

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

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

$$\widetilde{SJ} = \Big(\tfrac{1}{\rho^5}, \tfrac{1+3\rho^2+\rho^4}{5\rho^2}, \tfrac{1+\rho^2}{\rho^2(3-\rho^2)}, \tfrac{1}{\rho^2(2-\rho^2)}, \tfrac{1}{\rho^2(2-\rho^2)}, \tfrac{1+\rho^2}{\rho^2(3-\rho^2)}, \tfrac{1+\rho^2}{\rho^2(3-\rho^2)}, \tfrac{1}{\rho^2(2-\rho^2)}, \tfrac{1}{\rho^2(2-\rho^2)}, \tfrac{1+\rho^2}{\rho^2(3-\rho^2)}, \tfrac{1+\rho^2}{\rho^2(3-\rho^2)}, \tfrac{1}{\rho^2(2-\rho^2)}\Big).$$

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

$$\widetilde{VAR} = \Big(\tfrac{1}{\rho^{10}}, \tfrac{(1+3\rho^2+\rho^4)^2}{25\rho^4}, \tfrac{(1+\rho^2)^2}{\rho^4(3-\rho^2)^2}, \tfrac{1}{\rho^4(2-\rho^2)^2}, \tfrac{1}{\rho^4(2-\rho^2)^2}, \tfrac{(1+\rho^2)^2}{\rho^4(3-\rho^2)^2}, \tfrac{(1+\rho^2)^2}{\rho^4(3-\rho^2)^2}, \tfrac{1}{\rho^4(2-\rho^2)^2}, \tfrac{1}{\rho^4(2-\rho^2)^2}, \tfrac{(1+\rho^2)^2}{\rho^4(3-\rho^2)^2}, \tfrac{(1+\rho^2)^2}{\rho^4(3-\rho^2)^2}, \tfrac{1}{\rho^4(2-\rho^2)^2}\Big).$$

Let us denote $\chi = 1 - \rho^2$ and $\theta = 1 + 3\rho^2 + \rho^4$. The TPM for $DTMC(\overline{K})$ is

$$\widetilde{\mathbf{P}} = \begin{bmatrix}
1-\rho^5 & \rho^5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\chi^2}{\theta} & \frac{\rho^2\chi}{\theta} & \frac{\rho^4}{\theta} & \frac{\rho^4}{\theta} & \frac{\rho^2\chi}{\theta} & \frac{\rho^2\chi}{\theta} & \frac{\rho^4}{\theta} & \frac{\rho^4}{\theta} & \frac{\rho^2\chi}{\theta} & \frac{\rho^2\chi}{\theta} & \frac{\rho^4}{\theta} \\
0 & \frac{\rho^2\chi}{1+\rho^2} & \frac{\chi^2}{1+\rho^2} & \frac{\rho^2\chi}{1+\rho^2} & \frac{\rho^2\chi}{1+\rho^2} & \frac{\rho^4}{1+\rho^2} & \frac{\rho^4}{1+\rho^2} & 0 & 0 & 0 & 0 & 0 \\
0 & \rho^4 & \rho^2\chi & \chi^2 & 0 & \rho^2\chi & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \rho^4 & \rho^2\chi & 0 & \chi^2 & 0 & \rho^2\chi & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{\rho^2\chi}{1+\rho^2} & \frac{\rho^4}{1+\rho^2} & \frac{\rho^2\chi}{1+\rho^2} & 0 & \frac{\chi^2}{1+\rho^2} & 0 & \frac{\rho^2\chi}{1+\rho^2} & 0 & \frac{\rho^4}{1+\rho^2} & 0 & 0 \\
0 & \frac{\rho^2\chi}{1+\rho^2} & \frac{\rho^4}{1+\rho^2} & 0 & \frac{\rho^2\chi}{1+\rho^2} & 0 & \frac{\chi^2}{1+\rho^2} & 0 & \frac{\rho^2\chi}{1+\rho^2} & 0 & \frac{\rho^4}{1+\rho^2} & 0 \\
0 & \rho^4 & 0 & 0 & 0 & \rho^2\chi & 0 & \chi^2 & 0 & \rho^2\chi & 0 & 0 \\
0 & \rho^4 & 0 & 0 & 0 & 0 & \rho^2\chi & 0 & \chi^2 & 0 & \rho^2\chi & 0 \\
0 & \frac{\rho^2\chi}{1+\rho^2} & 0 & 0 & 0 & \frac{\rho^4}{1+\rho^2} & 0 & \frac{\rho^2\chi}{1+\rho^2} & 0 & \frac{\chi^2}{1+\rho^2} & \frac{\rho^4}{1+\rho^2} & \frac{\rho^2\chi}{1+\rho^2} \\
0 & \frac{\rho^2\chi}{1+\rho^2} & 0 & 0 & 0 & 0 & \frac{\rho^4}{1+\rho^2} & 0 & \frac{\rho^2\chi}{1+\rho^2} & \frac{\rho^4}{1+\rho^2} & \frac{\chi^2}{1+\rho^2} & \frac{\rho^2\chi}{1+\rho^2} \\
0 & \rho^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \rho^2\chi & \rho^2\chi & \chi^2
\end{bmatrix}.$$

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

$$\tilde{\psi} = \tfrac{1}{11+8\rho^2+\rho^4}(0, 1+3\rho^2+\rho^4, 1+\rho^2, 1, 1, 1+\rho^2, 1+\rho^2, 1, 1, 1+\rho^2, 1+\rho^2, 1).$$

We can now calculate the main performance indices.

- The average recurrence time in the state s_2, where all the forks are available, called the *average system run-through*, is $\frac{1}{\tilde{\psi}_2} = \frac{11+8\rho^2+\rho^4}{1+3\rho^2+\rho^4}$.
- Nobody eats in the state s_2. The *fraction of time when no philosophers dine* is $\tilde{\psi}_2 = \frac{1+3\rho^2+\rho^4}{11+8\rho^2+\rho^4}$.

  Only one philosopher eats in the states $s_3, s_6, s_7, s_{10}, s_{11}$. The *fraction of time when only one philosopher dines* is $\tilde{\psi}_3 + \tilde{\psi}_6 + \tilde{\psi}_7 + \tilde{\psi}_{10} + \tilde{\psi}_{11} = \frac{1+\rho^2}{11+8\rho^2+\rho^4} + \frac{1+\rho^2}{11+8\rho^2+\rho^4} + \frac{1+\rho^2}{11+8\rho^2+\rho^4} + \frac{1+\rho^2}{11+8\rho^2+\rho^4} + \frac{1+\rho^2}{11+8\rho^2+\rho^4} = \frac{5(1+\rho^2)}{11+8\rho^2+\rho^4}$.

  Two philosophers eat together in the states $s_4, s_5, s_8, s_9, s_{12}$. The *fraction of time when two philosophers dine* is $\tilde{\psi}_4 + \tilde{\psi}_5 + \tilde{\psi}_8 + \tilde{\psi}_9 + \tilde{\psi}_{12} = \frac{1}{11+8\rho^2+\rho^4} + \frac{1}{11+8\rho^2+\rho^4} + \frac{1}{11+8\rho^2+\rho^4} + \frac{1}{11+8\rho^2+\rho^4} + \frac{1}{11+8\rho^2+\rho^4} = \frac{5}{11+8\rho^2+\rho^4}$.

  The *relative fraction of time when two philosophers dine w.r.t. when only one philosopher dines* is $\frac{5}{11+8\rho^2+\rho^4} \cdot \frac{11+8\rho^2+\rho^4}{5(1+\rho^2)} = \frac{1}{1+\rho^2}$.

- The beginning of eating of first philosopher $(\{b_1\},\rho^2)$ is only possible from the states s_2, s_6, s_7. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of activities containing $(\{b_1\},\rho^2)$. Thus, the *steady-state probability of the beginning of eating of first philosopher* is $\tilde{\psi}_2 \sum_{\{\Gamma|(\{b_1\},\rho^2)\in\Gamma\}} PT(\Gamma,s_2) + \tilde{\psi}_6 \sum_{\{\Gamma|(\{b_1\},\rho^2)\in\Gamma\}} PT(\Gamma,s_6) + \tilde{\psi}_7 \sum_{\{\Gamma|(\{b_1\},\rho^2)\in\Gamma\}} PT(\Gamma,s_7) = \frac{1+3\rho^2+\rho^4}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} + \frac{\rho^4}{1+3\rho^2+\rho^4} + \frac{\rho^4}{1+3\rho^2+\rho^4}\right) + \frac{1+\rho^2}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2}\right) + \frac{1+\rho^2}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2}\right) = \frac{\rho^2(3+\rho^2)}{11+8\rho^2+\rho^4}$.

8.4 The abstract generalized system and its reduction

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

The static expression of the philosopher i $(1 \le i \le 4)$ is
$L_i = [(\{x_i\},\rho) * (((\{b,\widehat{y_i}\},\rho);(\{e,\widehat{z_i}\},\rho))[]\ ((\{y_{i+1}\},\rho);(\{z_{i+1}\},\rho))) * \mathsf{Stop}]$.

The static expression of the philosopher 5 is
$L_5 = [(\{a,\widehat{x_1},\widehat{x_2},\widehat{x_2},\widehat{x_4}\},\rho) * (((\{b,\widehat{y_5}\},\rho);(\{e,\widehat{z_5}\},\rho))[]((\{y_1\},\rho);(\{z_1\},\rho))) * \mathsf{Stop}]$.

The static expression of the abstract generalized dining philosophers system is
$L = (L_1\|L_2\|L_3\|L_4\|L_5)$ sy x_1 sy x_2 sy x_3 sy x_4 sy y_1 sy y_2 sy y_3 sy y_4 sy y_5 sy z_1 sy z_2 sy z_3 sy z_4 sy z_5 rs x_1 rs x_2 rs x_3 rs x_4 rs y_1 rs y_2 rs y_3 rs y_4 rs y_5 rs z_1 rs z_2 rs z_3 rs z_4 rs z_5.

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

The first performance index and the second group of the indices coincide for the standard and the abstract generalized systems. The following performance index is based on non-personalized viewpoint to the philosophers.

- The beginning of eating of a philosopher $(\{b\},\rho^2)$ is only possible from the states $\tilde{s}_2, \tilde{s}_3, \tilde{s}_6, \tilde{s}_7, \tilde{s}_{10}, \tilde{s}_{11}$. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of activities containing $(\{b\},\rho^2)$. Thus, the *steady-state probability of the beginning of eating of a philosopher* is $\tilde{\psi}_2 \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT(\Gamma,\tilde{s}_2) + \tilde{\psi}_3 \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT(\Gamma,\tilde{s}_3) + \tilde{\psi}_6 \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT(\Gamma,\tilde{s}_6) + \tilde{\psi}_7 \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT(\Gamma,\tilde{s}_7) + \tilde{\psi}_{10} \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT(\Gamma,\tilde{s}_{10}) + \tilde{\psi}_{11} \sum_{\{\Gamma|(\{b\},\rho^2)\in\Gamma\}} PT(\Gamma,\tilde{s}_{11}) = \frac{1+3\rho^2+\rho^4}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} + \frac{\rho^4}{1+3\rho^2+\rho^4} + \frac{\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} + \frac{\rho^4}{1+3\rho^2+\rho^4} + \frac{\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} + \frac{\rho^4}{1+3\rho^2+\rho^4} + \frac{\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} + \frac{\rho^4}{1+3\rho^2+\rho^4} + \frac{\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} + \frac{\rho^4}{1+3\rho^2+\rho^4}\right) + \frac{1+\rho^2}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2} + \frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2}\right) + \frac{1+\rho^2}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2} + \frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2}\right) + \frac{1+\rho^2}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2} + \frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2}\right) + \frac{1+\rho^2}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2} + \frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2}\right) + \frac{1+\rho^2}{11+8\rho^2+\rho^4}\left(\frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2} + \frac{\rho^2(1-\rho^2)}{1+\rho^2} + \frac{\rho^4}{1+\rho^2}\right) = \frac{15\rho^2}{11+8\rho^2+\rho^4}$.

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

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

$$\widetilde{SJ}' = \left(\frac{1}{\rho^5}, \frac{1+3\rho^2+\rho^4}{5\rho^2}, \frac{1+\rho^2}{3\rho^2(1-\rho^2)}, \frac{1}{\rho^2(2-\rho^2)}\right).$$

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

$$\widetilde{VAR}' = \left(\frac{1}{\rho^{10}}, \frac{(1+3\rho^2+\rho^4)^2}{25\rho^4}, \frac{(1+\rho^2)^2}{9\rho^4(1-\rho^2)^2}, \frac{1}{\rho^4(2-\rho^2)^2},\right).$$

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

$$\widetilde{\mathbf{P}}' = \begin{bmatrix} 1-\rho^5 & \rho^5 & 0 & 0 \\ 0 & \frac{(1-\rho^2)^2}{1+3\rho^2+\rho^4} & \frac{5\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} & \frac{5\rho^4}{1+3\rho^2+\rho^4} \\ 0 & \frac{\rho^2(1-\rho^2)}{1+\rho^2} & \frac{1-2\rho^2+3\rho^4}{1+\rho^2} & \frac{2\rho^2(1-\rho^2)}{1+\rho^2} \\ 0 & \rho^4 & 2\rho^2(1-\rho^2) & (1-\rho^2)^2 \end{bmatrix}.$$

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

$$\tilde{\psi}' = \left(0, \frac{1+3\rho^2+\rho^4}{11+8\rho^2+\rho^4}, \frac{5(1+\rho^2)}{11+8\rho^2+\rho^4}, \frac{5}{11+8\rho^2+\rho^4}\right).$$

We can now calculate the main performance indices.

- The average recurrence time in the state $\widetilde{\mathcal{K}}_2$, where all the forks are available, called the *average system run-through*, is $\frac{1}{\tilde{\psi}_2'} = \frac{11+8\rho^2+\rho^4}{1+3\rho^2+\rho^4}$.

- Nobody eats in the state $\widetilde{\mathcal{K}}_2$. The *fraction of time when no philosophers dine* is $\tilde{\psi}_2' = \frac{1+3\rho^2+\rho^4}{11+8\rho^2+\rho^4}$.

  Only one philosopher eats in the state $\widetilde{\mathcal{K}}_3$. The *fraction of time when only one philosopher dines* is $\tilde{\psi}_3' = \frac{5(1+\rho^2)}{11+8\rho^2+\rho^4}$.

  Two philosophers eat together in the state $\widetilde{\mathcal{K}}_4$. The *fraction of time when two philosophers dine* is $\tilde{\psi}_4' = \frac{5}{11+8\rho^2+\rho^4}$.

  The *relative fraction of time when two philosophers dine w.r.t. when only one philosopher dines* is $\frac{5}{11+8\rho^2+\rho^4} \cdot \frac{11+8\rho^2+\rho^4}{5(1+\rho^2)} = \frac{1}{1+\rho^2}$.

- The beginning of eating of a philosopher $\{b\}$ is only possible from the states $\widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}_3$. In each of the states, the beginning of eating probability is the sum of the execution probabilities for all multisets of multiactions containing $\{b\}$. Thus, the *steady-state probability of the beginning of eating of a philosopher* is $\tilde{\psi}_2' \sum_{\{A,\widetilde{\mathcal{K}} \mid \{b\}\in A,\ \widetilde{\mathcal{K}}_2 \xrightarrow{A} \widetilde{\mathcal{K}}\}} PM_A(\widetilde{\mathcal{K}}_2, \widetilde{\mathcal{K}}) + \tilde{\psi}_3' \sum_{\{A,\widetilde{\mathcal{K}} \mid \{b\}\in A,\ \widetilde{\mathcal{K}}_3 \xrightarrow{A} \widetilde{\mathcal{K}}\}} PM_A(\widetilde{\mathcal{K}}_3, \widetilde{\mathcal{K}}) =$ $\frac{1+3\rho^2+\rho^4}{11+8\rho^2+\rho^4}\left(\frac{5\rho^2(1-\rho^2)}{1+3\rho^2+\rho^4} + \frac{5\rho^4}{1+3\rho^2+\rho^4}\right) + \frac{5(1+\rho^2)}{11+8\rho^2+\rho^4}\left(\frac{2\rho^2(1-\rho^2)}{1+\rho^2} + \frac{2\rho^4}{1+\rho^2}\right) = \frac{15\rho^4}{11+8\rho^2+\rho^4}$.

Observe again that the performance indices are the same for the complete and the quotient abstract generalized dining philosophers systems. The explanation of this fact is just the same as that presented earlier for the complete and the quotient abstract dining philosophers systems.

Let us consider what is the effect of quantitative changes of the parameter ρ upon performance of the quotient abstract generalized dining philosophers system in its steady state. Remember that $\rho \in (0;1)$ is the probability of every multiaction of the system. The closer is ρ to 0, the less is the probability to execute some activities at every discrete time step, hence, the system will most probably *stand idle*. The closer is ρ to 1, the greater is the probability to execute some activities at every discrete time step, hence, the system will most probably *operate*.

Since $\tilde{\psi}_1' = 0$, only $\tilde{\psi}_2' = \frac{1+3\rho^2+\rho^4}{11+8\rho^2+\rho^4}$, $\tilde{\psi}_3' = \frac{5(1+\rho^2)}{11+8\rho^2+\rho^4}$, $\tilde{\psi}_4' = \frac{5}{11+8\rho^2+\rho^4}$ depend on ρ. In Figure 16, the graphs of $\tilde{\psi}_2'$, $\tilde{\psi}_3'$, $\tilde{\psi}_4'$ as functions of ρ are depicted. The diagrams for $\tilde{\psi}_2'$ and $\tilde{\psi}_4'$ are symmetric w.r.t. the constant probability $\frac{1}{4}$. One can see that, the more is value of ρ, the less is difference between $\tilde{\psi}_2'$ and $\tilde{\psi}_4'$ and the more is difference between $\tilde{\psi}_3'$ and $\tilde{\psi}_4'$. Notice that, however, we do not allow $\rho = 0$ or $\rho = 1$.

In Figure 17, the graph of the average system run-through, calculated as $\frac{1}{\tilde{\psi}_2'}$, as a function of ρ is depicted. One can see that the run-through tends to 11 when ρ approaches 0, whereas it tends to 4 when ρ approaches 1. To speed up the operation of the system, one should take the parameter ρ closer to 1.

The fraction of time when no philosophers dine, calculated as $\tilde{\psi}_2'$, tends to $\frac{1}{11}$ when ρ approaches 0, whereas it tends to $\frac{1}{4}$ when ρ approaches 1. The fraction of time when only one philosopher dines, calculated as $\tilde{\psi}_3'$, tends to $\frac{5}{11}$ when ρ approaches 0, whereas it tends to $\frac{1}{2}$ when ρ approaches 1. The fraction of time when two philosophers dine, calculated as $\tilde{\psi}_4'$, tends to $\frac{5}{11}$ when ρ approaches 0, whereas it tends to $\frac{1}{4}$ when ρ approaches 1.

The first graph in Figure 18 represents the relative fraction of time when two philosophers dine w.r.t. when only one philosopher dines, calculated as $\frac{\tilde{\psi}_4'}{\tilde{\psi}_3'}$, as a function of ρ. One can see that the relative fraction tends to 1 when ρ approaches 0, whereas it tends to $\frac{1}{2}$ when ρ approaches 1. To increase the mentioned relative fraction, one should take the parameter ρ closer to 0.

The second graph in Figure 18 represents the steady-state probability of the beginning of eating of a philosopher, calculated as $\tilde{\psi}_2'\widetilde{\Sigma}_2' + \tilde{\psi}_3'\widetilde{\Sigma}_3'$, where $\widetilde{\Sigma}_i' = \sum_{\{A,\widetilde{\mathcal{K}} \mid \{b\}\in A,\ \widetilde{\mathcal{K}}_i \xrightarrow{A} \widetilde{\mathcal{K}}\}} PM_A(\widetilde{\mathcal{K}}_i, \widetilde{\mathcal{K}})$, $i \in \{2,3\}$, as a function of ρ. One can see that the probability tends to 0 when ρ approaches 0 and it tends to $\frac{3}{4}$, when ρ approaches 1. To increase the mentioned probability, one should take the parameter ρ closer to 1.

Since $\tilde{\psi}_4' - \tilde{\psi}_2' = \frac{4-3\rho^2-\rho^4}{11+8\rho^2+\rho^4}$, the difference tends to $\frac{4}{11}$ when ρ approaches 0, whereas it tends to 0 when ρ approaches 1. The difference can be treated as that between the fractions of time when two and when no philosophers dine. Since $\tilde{\psi}_3' - \tilde{\psi}_4' = \frac{5\rho^2}{11+8\rho^2+\rho^4}$, the difference tends to 0 when ρ approaches 0, whereas it tends

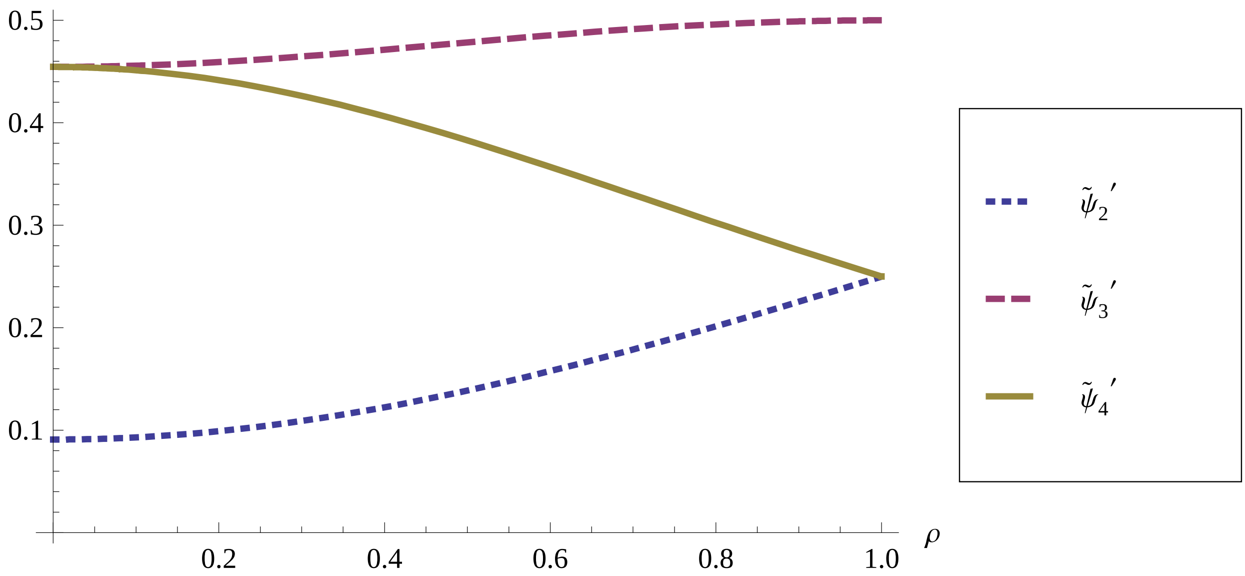

Figure 16: Steady-state probabilities $\tilde{\psi}_2'$, $\tilde{\psi}_3'$, $\tilde{\psi}_4'$ as functions of the parameter ρ

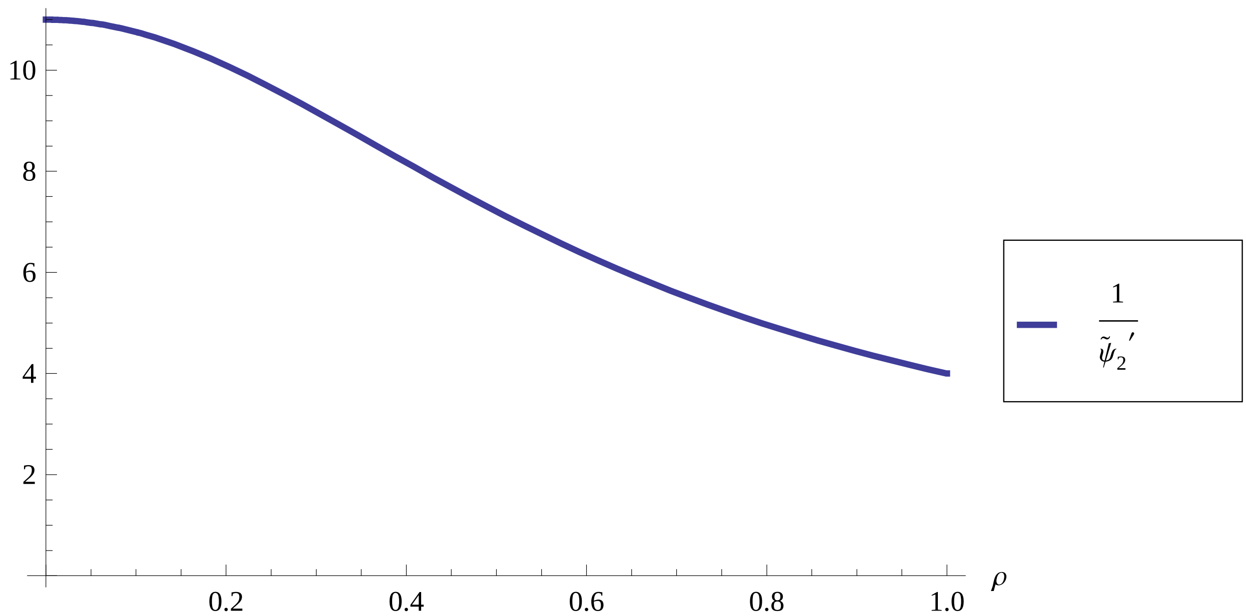

Figure 17: Average system run-through $\frac{1}{\tilde{\psi}_2'}$ as a function of the parameter ρ

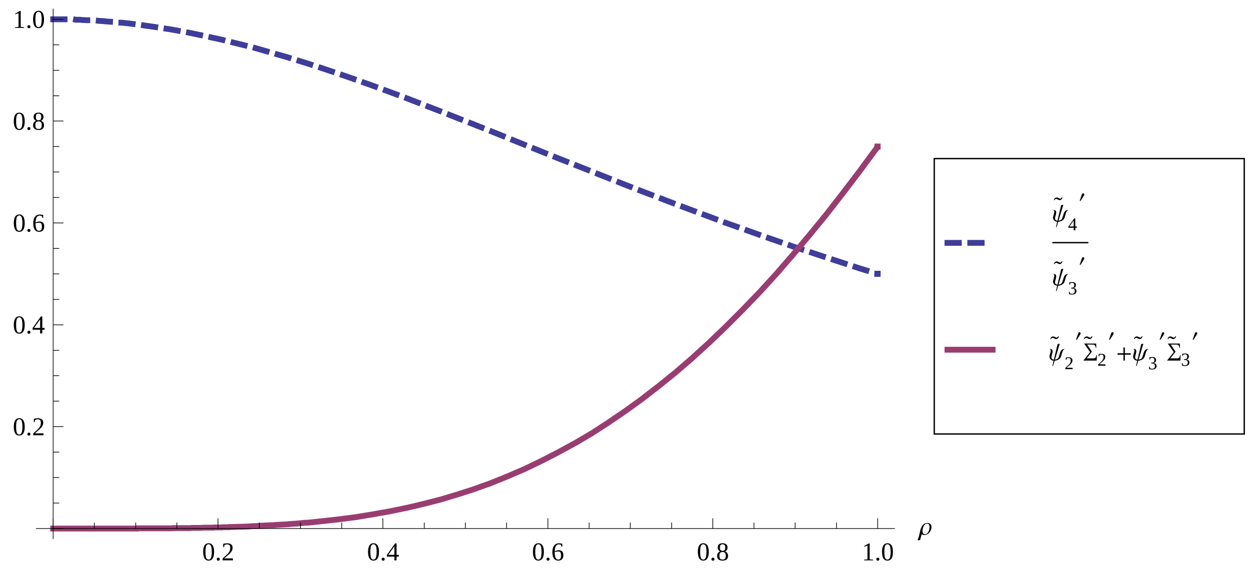

Figure 18: Some performance indices as functions of the parameter ρ

to $\frac{1}{4}$ when ρ approaches 1. The difference can be treated as the that between the fractions of time when one and when two philosophers dine.

From the performance viewpoint, it is more interesting is to consider the expression $\tilde{\psi}'_3+\tilde{\psi}'_4-\tilde{\psi}'_2 = \frac{9+2\rho^2-\rho^4}{11+8\rho^2+\rho^4}$. In Figure 19, the graph of $\tilde{\psi}'_3 + \tilde{\psi}'_4 - \tilde{\psi}'_2$ as a function of ρ is depicted. The value of the expression tends to $\frac{9}{11}$ when ρ approaches 0, whereas it tends to $\frac{1}{2}$ when ρ approaches 1. The value can be interpreted as the difference between the fractions of time when some (one or two) and when no philosophers dine.

Thus, when ρ is closer to 0, more time is spent for eating and less time remains for thinking, i.e., *dining is preferred*. In this case, the dining time fractions for one and two philosophers approach the same value $\frac{5}{11}$ (the relative time fraction tends to 1). When ρ is closer to 1, the situation is symmetric, i.e., *thinking is preferred*. In this case, the dining time fraction of one philosopher approaches its maximum $\frac{1}{2}$, whereas the dining time fraction of two philosophers approaches its minimum $\frac{1}{4}$ (the relative time fraction tends to $\frac{1}{2}$).

9 Discussion

Let us compare dtsPBC with the classical SPAs MTIPP, PEPA and EMPA in detail. In dtsPBC, every activity is a pair consisting of the multiaction (not just an action, as in the classical SPAs) and its probability (not the rate, as in the classical SPAs) to be executed under condition that no other multiaction can occur at the current discrete time moment. dtsPBC has the sequence operation in contrast to the prefix one in the classical SPAs. One can combine arbitrary expressions with the sequence operator, i.e., it is more flexible than the prefix one, where the first argument should be a single activity. The choice operation in dtsPBC is analogous to that in MTIPP and PEPA, as well as to the alternative composition in EMPA, in the sense that the choice is probabilistic, but a discrete probability function is used in dtsPBC, unlike continuous ones in the classical calculi. Concurrency and synchronization in dtsPBC are different operations (this feature is inherited from PBC) in contrast to the classical SPAs, where parallel composition (combinator) has a synchronization capability. Relabeling in dtsPBC is analogous to that in EMPA, but it is additionally extended to conjugated actions. The restriction operation in dtsPBC differs from hiding in PEPA and functional abstraction in EMPA, where the hidden actions are labeled with a symbol of "silent" action τ. In dtsPBC, restriction by an action means that for a given expression any process behaviour containing the action or its conjugate is not allowed. The synchronization on an elementary action collects all the pairs consisting of this elementary action and its conjugate which are contained in the multiactions from the synchronized activities. The operation produces new activities such that the first element of every resulting activity is the union of the multiactions from which all the mentioned pairs of conjugated actions are removed, and the second element is the product of the probabilities of the activities involved in the synchronization. This differs from the way synchronization is applied in the classical SPAs where it is accomplished over identical action names, and every resulting activity consist of the same action name and the rate calculated via some expression (including sums, minimums and products) on the

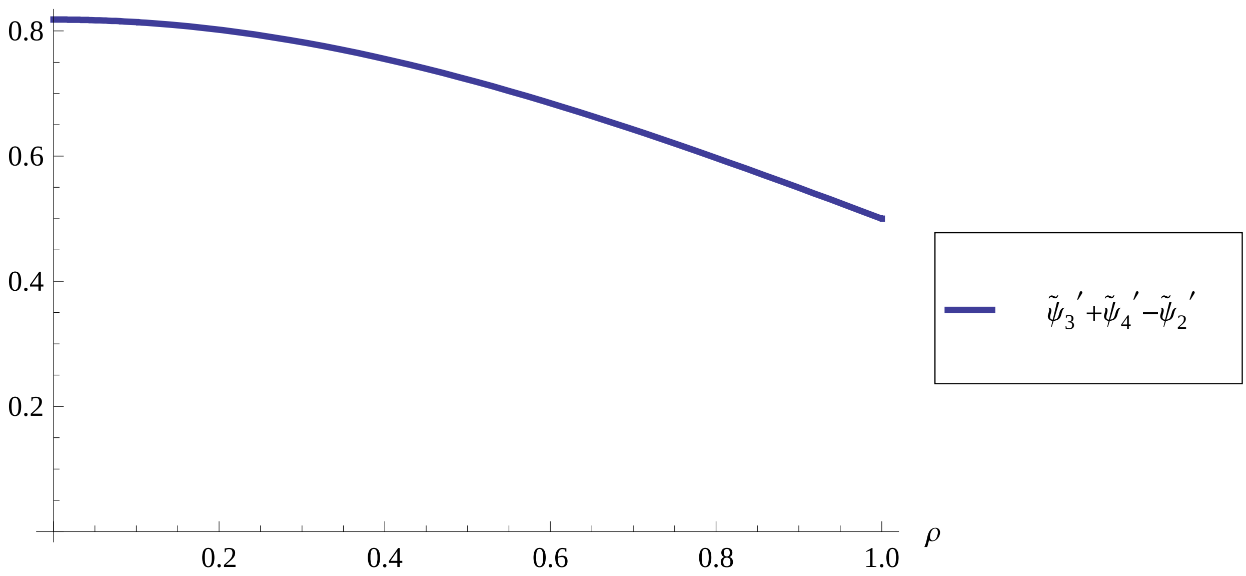

Figure 19: Expression $\tilde{\psi}_3' + \tilde{\psi}_4' - \tilde{\psi}_2'$ as a function of the parameter ρ

rates of the initial activities, such as the apparent rate in PEPA. dtsPBC has no recursion operation or recursive definitions, but it includes the iteration operation to specify infinite looping behaviour with the explicitly defined start and termination. dtsPBC has a discrete time semantics, and time delays in the states are geometrically distributed, unlike the classical SPAs with continuous time semantics and exponentially distributed activity delays. As a consequence, the semantics of dtsPBC is the step one in contrast to the interleaving semantics of the classical SPAs. The performance can be investigated based on the discrete time Markov chain (DTMC) extracted from the labeled probabilistic transition system associated with each expression of dtsPBC. In the classical SPAs, continuous time Markov chains (CTMCs) are used for performance evaluation. dtsPBC has a denotational semantics in terms of LDTSPNs from which the corresponding DTMCs can be derived as well.

Only a few non-interleaving SPAs were proposed among non-Markovian ones [18]. The semantics of all Markovian calculi is interleaving and their action delays have exponential distribution, which is the only continuous time probability distribution with memoryless (Markovian) property. In [8], generalized stochastic process algebra (GSPA) was introduced. It has a true-concurrent denotational semantics in terms of generalized stochastic event structures (GSESs) with non-Markovian stochastic delays of events. In that paper, no operational semantics or performance evaluation methods for GSPA were presented. Later, in [19], generalized semi-Markov processes (GSMPs) were extracted from GSESs to analyze performance. In [33], stochastic π-calculus (Sπ) with general continuous distributions of activity delays was defined. It has a proved operational semantics with transitions labeled by encodings of their deduction trees. No well-established underlying performance model for Sπ was described. In [2], generalized semi-Markovian process algebra (GSMPA) was developed with ST-operational semantics and non-Markovian action delays. The performance analysis in GSMPA is accomplished via GSMPs.

Again, the first fundamental difference between dtsPBC and the calculi GSPA, Sπ and GSMPA is that dtsPBC is based on PBC, whereas GSPA is an extension of process algebra (PA) from [8], Sπ extends π-calculus [25] and GSMPA is an enrichment of EMPA. The second significant difference is that geometrically distributed delays are associated with process states in dtsPBC, unlike generally distributed delays assigned to events in GSPA or to activities in Sπ and GSMPA. As a consequence, dtsPBC has a discrete time operational semantics allowing for concurrent execution of activities in steps. GSPA has no operational semantics while Sπ and GSMPA have continuous time ones. In continuous time semantics, concurrency is simulated by interleaving, since simultaneous occurrence of any two events has zero probability according to the properties of continuous time probability distributions. Therefore, interleaving transitions are often annotated with an additional information to keep concurrency data. The transition labels in the operational semantics of Sπ encode the action causality information and allow one to derive the enabling relations and the firing distributions of concurrent transitions from the transition sequences. At the same time, abstracting from stochastic delays leads to the classical early interleaving semantics of π-calculus. ST-operational semantics of GSMPA is based on decorated transition systems governed by transition rules with rather complex preconditions. There are two types of transitions: the choice (action beginning) and the termination (action ending) ones. The choice transitions are labeled by weights of single actions chosen for execution while the termination transitions have no labels. Step transition systems could be retrieved from the decorated (interleaving) ones. dtsPBC has an SPNs-based denotational semantics. In comparison with event structures, PNs are more expressive and visually tractable formalism capable of finitely specifying an infinite behaviour. Recursion in GSPA produces infinite GSESs while dtsPBC has iteration operation with a finite SPN semantics. An identification of infinite GSESs that can be finitely represented in GSPA was left for a future research.

In [29, 30], an SPA called theory of communicating processes with discrete stochastic time (TCP^{dts}) was introduced. Its actions have a (deterministic) discrete real time delays (including zero time delays) or stochastic time delays. The algebra generalizes real-time processes to discrete stochastic time ones by applying real-time properties to stochastic time and imposing race condition to real time semantics. TCP^{dts} has an interleaving operational semantics in terms of stochastic transition systems. The performance is analyzed via discrete time probabilistic reward graphs which are essentially the reward transition systems with probabilistic states having finite number of outgoing probabilistic transitions and timed states having a single outgoing timed transition. The mentioned graphs can be transformed by unfolding or geometrization into discrete time Markov reward chains appropriate for transient or long-run (stationary) analysis.

The first difference between dtsPBC and TCP^{dts} is that dtsPBC is based on PBC, but TCP^{dts} is an extension of algebra of communicating processes (ACP) [7]. The second difference is that geometrically distributed delays are associated with process states in dtsPBC, unlike deterministic or generally distributed stochastic delays of actions in TCP^{dts}. In spite of the discrete time approach, operational semantics of TCP^{dts} is still interleaving, unlike that of dtsPBC. In addition, no denotational semantics was defined for TCP^{dts}.

Thus, the multiaction labels, the set of flexible and powerful operations, as well as a step operational and a Petri net denotational semantics allowing for concurrent execution of activities (or transitions) are the main advantages of dtsPBC. The salient point of dtsPBC is a combination of discrete stochastic time and step semantics in an SPA.

10 Conclusion

In this paper, within the context of dtsPBC with iteration, we have defined the stochastic algebraic equivalences having natural net analogues on LDTSPNs. The diagram of interrelations for the algebraic equivalences has been constructed. We have explained how one can reduce transition systems and DTMCs, as well as expressions and dts-boxes modulo the stochastic equivalences. We have investigated which of the equivalences we proposed guarantee identity of the stationary behaviour. We have proved that the weakest among the relations we have considered, step stochastic bisimulation equivalence, has this property. A case study of the dining philosophers system has been presented as an example of modeling, performance evaluation and performance preserving reduction in the framework of the calculus. The advantage of our framework is twofold. First, one can specify in it concurrent composition and synchronization of (multi)actions, whereas this is not possible in classical Markov chains. Second, algebraic formulas represent processes in a more compact way than PNs and allow one to apply syntactic transformations and comparisons.

In the future, we plan to provide the stochastic equivalences with a logical characterization via probabilistic modal logics. Abstracting from the silent activities in the definitions of the equivalences, i.e., from the activities with empty multiaction part, is the next research direction. The main point here is that we should collect probabilities during such abstractions from an internal activity. Moreover, we plan to extend dtsPBC with recursion to enhance the specification power of the calculus.

Acknowledgements I would like to thank Eike Best for the support and encouraging discussions during my research stay at the University of Oldenburg. I am also grateful to Peter Buchholz for inviting me to present my results at the Technical University of Dortmund, and to Falko Bause for his interest to my work and the qualified consideration. Many thanks to anonymous referees for their valuable advices on improvement of the previous version of this paper.

References

[1] Autant, C., Schnoebelen, Ph.: Place bisimulations in Petri nets. Lect. Notes Comp. Sci. **616**, 45–61 (1992).

[2] Bravetti, M., Bernardo, M., Gorrieri, R.: Towards performance evaluation with general distributions in process algebras. Lect. Notes Comp. Sci. **1466**, 405-422 (2008), http://www.cs.unibo.it/~bravetti/papers/concur98.ps.

[3] Best, E., Devillers, R., Hall, J.G.: The box calculus: a new causal algebra with multi-label communication. Lect. Notes Comp. Sci. **609**, 21–69 (1992)

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

[5] Bernardo, M.: A survey of Markovian behavioral equivalences. Lect. Notes Comp. Sci. **4486**, 180–219 (2007)

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

[7] Bergstra, J.A., Klop, J.W.: Algebra of communicating processes with abstraction. Theor. Comput. Sci. **37**, 77–121 (1985)

[8] Brinksma, E., Katoen, J.-P., Langerak, R., Latella, D.: A stochastic causality-based process algebra. Comp. J. **38 (7)**, 552-565 (1995), http://eprints.eemcs.utwente.nl/6387/01/552.pdf.

[9] Best, E., Koutny, M.: A refined view of the box algebra. Lect. Notes Comp. Sci. **935**, 1–20 (1995)

[10] Brinksma, E., Hermanns, H.: Process algebra and Markov chains. Lect. Notes Comp. Sci. **2090**, 183–231 (2001)

[11] Buchholz, P.: A notion of equivalence for stochastic Petri nets. Lect. Notes Comp. Sci. **935**, 161–180 (1995)

[12] Buchholz, P.: Iterative decomposition and aggregation of labeled GSPNs. Lect. Notes Comp. Sci. **1420**, 226–245 (1998)

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

[14] van Glabbeek, R.J., Smolka, S.A., Steffen, B.: Reactive, generative, and stratified models of probabilistic processes. Information and Computation **121**(1), 59–80 (1995), http://boole.stanford.edu/pub/prob.ps.gz.

[15] Hillston, J.: A compositional approach to performance modelling. Cambridge University Press, Great Britain (1996)

[16] Hermanns, H., Rettelbach, M.: Syntax, semantics, equivalences and axioms for MTIPP. Proceedings of 2^{nd} Workshop on Process Algebras and Performance Modelling (Herzog U., Rettelbach M., eds.), Arbeitsberichte des IMMD **27**, 71–88, University of Erlangen, Germany (1994)

[17] Jou, C.-C., Smolka, S.A.: Equivalences, congruences and complete axiomatizations for probabilistic processes. Lect. Notes Comp. Sci. **458**, 367–383 (1990)

[18] Katoen, J.-P., D'Argenio, P.R.: General distributions in process algebra. Lect. Notes Comp. Sci. **2090**, 375-429 (2001)

[19] Katoen, J.-P., Brinksma, E., Latella, D., Langerak, R.: Stochastic simulation of event structures. Proceedings of 4^{th} International Workshop on Process Algebra and Performance Modelling (PAPM'96) (M. Ribaudo, ed.), 21-40, CLUT Press, Torino, Italy (1996), http://eprints.eemcs.utwente.nl/6487/01/263˙KLLB96b.pdf.

[20] Koutny, M.: A compositional model of time Petri nets. Lect. Notes Comp. Sci. **1825**, 303–322 (2000)

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

[22] Marroquín, O., de-Frutos, D.: Extending the Petri box calculus with time. Lect. Notes Comp. Sci. **2075**, 303–322 (2001)

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

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

[25] Milner, R.A.J., Parrow, J., Walker, D.: A calculus of mobile processes (I and II). Information and Computation **100**(1), 1–77 (1992)

[26] Macià, H., Valero, V., Cazorla, D., Cuartero, F.: Introducing the iteration in sPBC. Proceedings of the 24^{th} International Conference on Formal Techniques for Networked and Distributed Systems - 04 (FORTE'04), Madrid, Spain, Lect. Notes Comp. Sci. **3235**, 292–308 (2004)

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

[28] Macià, H., Valero, V., de-Frutos, D.: sPBC: a Markovian extension of finite Petri box calculus. Proceedings of 9^{th} IEEE International Workshop on Petri Nets and Performance Models (PNPM'01), 207–216, Aachen, Germany, IEEE Computer Society Press (2001)

[29] Markovski, J., de Vink, E.P.: Extending timed process algebra with discrete stochastic time. Proceedings of 12^{th} International Conference on Algebraic Methodology and Software Technology - 08 (AMAST'08), Urbana, IL, USA, Lect. Notes Comp. Sci. **5140**, 268–283 (2008)

[30] Markovski, J., de Vink, E.P.: Performance evaluation of distributed systems based on a discrete real- and stochastic-time process algebra. Fundamenta Informaticae **95(1)**, 157–186, IOS Press, Amsterdam, The Netherlands (2009)

[31] Niaouris, A.: An algebra of Petri nets with arc-based time restrictions. Lect. Notes Comp. Sci. **3407**, 447–462 (2005)

[32] Peterson, J.L.: Petri net theory and modeling of systems. Prentice-Hall (1981)

[33] Priami, C.: Stochastic π-calculus with general distributions. Proceedings of 4^{th} International Workshop on Process Algebra and Performance Modelling (PAPM'96) (M. Ribaudo, ed.), 41–57, CLUT Press, Torino, Italy (1996)

[34] Paige, R., Tarjan, R.E.: Three partition refinement algorithms. SIAM J. Comput. **16(6)**, 973–989 (1987)

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

[36] Tarasyuk, I.V.: Iteration in discrete time stochastic Petri box calculus. Bulletin of the Novosibirsk Computing Center, Series Computer Science, IIS Special Issue **24**, 129–148, NCC Publisher, Novosibirsk (2006), http://db.iis.nsk.su/persons/itar/dtsitncc.pdf.

[37] Tarasyuk, I.V.: Stochastic Petri box calculus with discrete time. Fundamenta Informaticae **76**(1–2), 189–218, IOS Press, Amsterdam, The Netherlands (2007), http://db.iis.nsk.su/persons/itar/dtspbcfi.pdf.

[38] Tarasyuk, I.V.: Investigating equivalence relations in dtsPBC. Berichte aus dem Department für Informatik **5/08**, 57 pages, Carl von Ossietzky Universität Oldenburg, Germany (October 2008), http://db.iis.nsk.su/persons/itar/dtspbcit˙cov.pdf.

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

A Proofs

A.1 Proof of Proposition 5.1

Like it has been done for strong equivalence in Proposition 8.2.1 from [15], we shall prove the following fact about step stochastic bisimulation. Let us have $\forall j \in \mathcal{J}\ \mathcal{R}_j : G \underline{\leftrightarrow}_{ss} G'$ for some index set $\mathcal{J}$. Then the transitive closure of the union of all relations $\mathcal{R} = (\cup_{j\in\mathcal{J}}\mathcal{R}_j)^*$ is also an equivalence and $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$.

Since $\forall j \in \mathcal{J}\ \mathcal{R}_j$ is an equivalence, by definition of $\mathcal{R}$, we get that $\mathcal{R}$ is also an equivalence.

Let $j \in \mathcal{J}$, then, by definition of $\mathcal{R}$, $(s_1, s_2) \in \mathcal{R}_j$ implies $(s_1, s_2) \in \mathcal{R}$. Hence, $\forall \mathcal{H}_{jk} \in (DR(G) \cup DR(G'))/_{\mathcal{R}_j}\ \exists \mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}\ \mathcal{H}_{jk} \subseteq \mathcal{H}$. Moreover, $\exists \mathcal{J}'\ \mathcal{H} = \cup_{k\in\mathcal{J}'}\mathcal{H}_{jk}$.

We denote $\mathcal{R}(n) = (\cup_{j\in\mathcal{J}}\mathcal{R}_j)^n$. Let $(s_1, s_2) \in \mathcal{R}$, then, by definition of $\mathcal{R}$, $\exists n > 0\ (s_1, s_2) \in \mathcal{R}(n)$. We shall prove that $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$ by induction on n.

It is clear that $\forall j \in \mathcal{J}\ \mathcal{R}_j : G\underline{\leftrightarrow}_{ss}G'$ implies $\forall j \in \mathcal{J}\ ([G]_\approx, [G']_\approx) \in \mathcal{R}_j$ and we have $([G]_\approx, [G']_\approx) \in \mathcal{R}$ by definition of $\mathcal{R}$.

It remains to prove that $(s_1, s_2) \in \mathcal{R}$ implies $\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_\mathcal{R}\ \forall A \in I\!N_f^\mathcal{L}\ PM_A(s_1, \mathcal{H}) = PM_A(s_2, \mathcal{H})$.

- $n = 1$

  In this case, $(s_1, s_2) \in \mathcal{R}$ implies $\exists j \in \mathcal{J}\ (s_1, s_2) \in \mathcal{R}_j$. Since $\mathcal{R}_j : G\underline{\leftrightarrow}_{ss}G'$, we get $\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_\mathcal{R}\ \forall A \in I\!N_f^\mathcal{L}$

  $$PM_A(s_1, \mathcal{H}) = \sum_{k \in \mathcal{J}'} PM_A(s_1, \mathcal{H}_{jk}) = \sum_{k \in \mathcal{J}'} PM_A(s_2, \mathcal{H}_{jk}) = PM_A(s_2, \mathcal{H}).$$

- $n \to n + 1$

  Suppose that $\forall m \le n\ (s_1, s_2) \in \mathcal{R}(m)$ implies $\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_\mathcal{R}\ \forall A \in I\!N_f^\mathcal{L}\ PM_A(s_1, \mathcal{H}) = PM_A(s_2, \mathcal{H})$.

  Then $(s_1, s_2) \in \mathcal{R}(n + 1)$ implies $\exists j \in \mathcal{J}\ (s_1, s_2) \in \mathcal{R}_j \circ \mathcal{R}(n)$, i.e., $\exists s_3 \in (DR(G) \cup DR(G'))$ such that $(s_1, s_3) \in \mathcal{R}_j$ and $(s_3, s_2) \in \mathcal{R}(n)$.

  Then, like for the case $n = 1$, we get $PM_A(s_1, \mathcal{H}) = PM_A(s_3, \mathcal{H})$. By the induction hypothesis, $PM_A(s_3, \mathcal{H}) = PM_A(s_2, \mathcal{H})$. Thus, $\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_\mathcal{R}\ \forall A \in I\!N_f^\mathcal{L}$

  $$PM_A(s_1, \mathcal{H}) = PM_A(s_3, \mathcal{H}) = PM_A(s_2, \mathcal{H}).$$

By definition, $\mathcal{R}_{ss}(G, G')$, is at least as large as the largest step stochastic bisimulation between G and G'. It follows from mentioned above that $\mathcal{R}_{ss}(G, G') : G\underline{\leftrightarrow}_{ss}G'$. □

A.2 Proof of Proposition 7.1

It is sufficient to prove the statement of the proposition for transient PMFs only, since $\psi = \lim_{k\to\infty} \psi[k]$ and $\psi' = \lim_{k\to\infty} \psi'[k]$. We proceed by induction on k.

- $k = 0$

  Note that the only nonzero values of the initial PMFs of $DTMC(G)$ and $DTMC(G')$ are $\psi[0]([G]_\approx)$ and $\psi[0]([G']_\approx)$. The only equivalence class containing $[G]_\approx$ or $[G']_\approx$ is $\mathcal{H}_0 = \{[G]_\approx, [G']_\approx\}$. Thus, $\sum_{s\in\mathcal{H}_0\cap DR(G)} \psi[0](s) = \psi[0]([G]_\approx) = 1 = \psi'[0]([G']_\approx) = \sum_{s'\in\mathcal{H}_0\cap DR(G')} \psi'[0](s')$.

  As for other equivalence classes, $\forall \mathcal{H} \in ((DR(G) \cup DR(G'))/_\mathcal{R}) \setminus \mathcal{H}_0$ we have $\sum_{s\in\mathcal{H}\cap DR(G)} \psi[0](s) = 0 = \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'[0](s')$.

- $k \to k + 1$

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

  By induction hypothesis, $\sum_{s\in\mathcal{H}\cap DR(G)} \psi[k](s) = \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'[k](s')$. Further,
  $\sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} \psi[k+1](\tilde{s}) = \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} \sum_{s\in DR(G)} \psi[k](s) PM(s, \tilde{s}) =$
  $\sum_{s\in DR(G)} \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} \psi[k](s) PM(s, \tilde{s}) = \sum_{s\in DR(G)} \psi[k](s) \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} PM(s, \tilde{s}) =$
  $\sum_\mathcal{H} \sum_{s\in\mathcal{H}\cap DR(G)} \psi[k](s) \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} PM(s, \tilde{s}) =$
  $\sum_\mathcal{H} \sum_{s\in\mathcal{H}\cap DR(G)} \psi[k](s) \sum_{\tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)} \sum_{\{\Gamma \mid s \stackrel{\Gamma}{\to} \tilde{s}\}} PT(\Gamma, s) =$
  $\sum_\mathcal{H} \sum_{s\in\mathcal{H}\cap DR(G)} \psi[k](s) \sum_{\{\Gamma \mid \exists \tilde{s}\in\widetilde{\mathcal{H}}\cap DR(G)\ s \stackrel{\Gamma}{\to} \tilde{s}\}} PT(\Gamma, s) =$
  $\sum_\mathcal{H} \sum_{s\in\mathcal{H}\cap DR(G)} \psi[k](s) PM(s, \widetilde{\mathcal{H}}) = \sum_\mathcal{H} \sum_{s\in\mathcal{H}\cap DR(G)} \psi[k](s) PM(\mathcal{H}, \widetilde{\mathcal{H}}) =$
  $\sum_\mathcal{H} PM(\mathcal{H}, \widetilde{\mathcal{H}}) \sum_{s\in\mathcal{H}\cap DR(G)} \psi[k](s) = \sum_\mathcal{H} PM(\mathcal{H}, \widetilde{\mathcal{H}}) \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'[k](s') =$
  $\sum_\mathcal{H} \sum_{s'\in\mathcal{H}\cap DR(G')} \psi'[k](s') PM(\mathcal{H}, \widetilde{\mathcal{H}}) = \sum_\mathcal{H} \sum_{s'\in\mathcal{H}'\cap DR(G')} \psi'[k](s') PM(s', \widetilde{\mathcal{H}}) =$

$\sum_{\mathcal{H}} \sum_{s' \in \mathcal{H} \cap DR(G')} \psi'[k](s') \sum_{\{\Gamma \mid \exists \tilde{s}' \in \widetilde{\mathcal{H}} \cap DR(G')\ s' \stackrel{\Gamma}{\to} \tilde{s}'\}} PT(\Gamma, s') =$
$\sum_{\mathcal{H}} \sum_{s' \in \mathcal{H} \cap DR(G')} \psi'[k](s') \sum_{\tilde{s}' \in \widetilde{\mathcal{H}} \cap DR(G')} \sum_{\{\Gamma \mid \exists \tilde{s}'\ s' \stackrel{\Gamma}{\to} \tilde{s}'\}} PT(\Gamma, s') =$
$\sum_{\mathcal{H}} \sum_{s' \in \mathcal{H} \cap DR(G')} \psi'[k](s') \sum_{\tilde{s}' \in \widetilde{\mathcal{H}} \cap DR(G')} PM(s', \tilde{s}') = \sum_{s' \in DR(G')} \psi'[k](s') \sum_{\tilde{s}' \in \widetilde{\mathcal{H}} \cap DR(G')} PM(s', \tilde{s}') =$

$\sum_{s' \in DR(G')} \sum_{\tilde{s}' \in \widetilde{\mathcal{H}} \cap DR(G')} \psi'[k](s') PM(s', \tilde{s}') = \sum_{\tilde{s}' \in \widetilde{\mathcal{H}} \cap DR(G')} \sum_{s' \in DR(G')} \psi'[k](s') PM(s', \tilde{s}') =$
$\sum_{\tilde{s}' \in \widetilde{\mathcal{H}} \cap DR(G')} \psi'[k+1](\tilde{s}')$. ◻

A.3 Proof of Theorem 7.1

Let $\mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$ and $s, \bar{s} \in \mathcal{H}$. We have $\forall \widetilde{\mathcal{H}} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}\ \forall A \in I\!N_f^{\mathcal{L}}\ s \stackrel{A}{\to}_{\mathcal{P}} \widetilde{\mathcal{H}} \Leftrightarrow \bar{s} \stackrel{A}{\to}_{\mathcal{P}} \widetilde{\mathcal{H}}$. The previous statement is valid for all $s, \bar{s} \in \mathcal{H}$, hence, we can rewrite it as $\mathcal{H} \stackrel{A}{\to}_{\mathcal{P}} \widetilde{\mathcal{H}}$ and denote $PM_A(\mathcal{H}, \widetilde{\mathcal{H}}) = PM_A(s, \widetilde{\mathcal{H}}) = PM_A(\bar{s}, \widetilde{\mathcal{H}})$. Note that transitions from the states of $DR(G)$ always lead to those from the same set, hence, $\forall s \in DR(G)\ PM_A(s, \widetilde{\mathcal{H}}) = PM_A(s, \widetilde{\mathcal{H}} \cap DR(G))$. The same is true for $DR(G')$.

Let $\Sigma = A_1 \cdots A_n$ be a step trace of G and G'. We have $\exists \mathcal{H}_0, \ldots, \exists \mathcal{H}_n \in (DR(G) \cup DR(G'))/_{\mathcal{R}}\ \mathcal{H}_0 \stackrel{A_1}{\to}_{\mathcal{P}_1} \mathcal{H}_1 \stackrel{A_2}{\to}_{\mathcal{P}_2} \cdots \stackrel{A_n}{\to}_{\mathcal{P}_n} \mathcal{H}_n$. Now we prove that the sum of probabilities of all the paths starting in every $s_0 \in \mathcal{H}_0$ and going through the states from $\mathcal{H}_1, \ldots, \mathcal{H}_n$ is equal to the product of $\mathcal{P}_1, \ldots, \mathcal{P}_n$:

$$\sum_{\{\Gamma_1, \ldots, \Gamma_n \mid s_0 \stackrel{\Gamma_1}{\to} \cdots \stackrel{\Gamma_n}{\to} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}) = \prod_{i=1}^{n} PM_{A_i}(\mathcal{H}_{i-1}, \mathcal{H}_i).$$

We prove this equality by induction on the step trace length n.

- $n = 1$
  $\sum_{\{\Gamma_1 \mid s_0 \stackrel{\Gamma_1}{\to} s_1,\ \mathcal{L}(\Gamma_1) = A_1,\ s_1 \in \mathcal{H}_1\}} PT(\Gamma_1, s_0) = PM_{A_1}(s_0, \mathcal{H}_1) = PM_{A_1}(\mathcal{H}_0, \mathcal{H}_1)$.
- $n \to n+1$
  $\sum_{\{\Gamma_1, \ldots, \Gamma_n, \Gamma_{n+1} \mid s_0 \stackrel{\Gamma_1}{\to} \cdots \stackrel{\Gamma_n}{\to} s_n \stackrel{\Gamma_{n+1}}{\to} s_{n+1},\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n+1)\}} \prod_{i=1}^{n+1} PT(\Gamma_i, s_{i-1}) =$
  $\sum_{\{\Gamma_{n+1} \mid s_n \stackrel{\Gamma_{n+1}}{\to} s_{n+1},\ \mathcal{L}(\Gamma_{n+1}) = A_{n+1},\ s_n \in \mathcal{H}_n,\ s_{n+1} \in \mathcal{H}_{n+1}\}}$
  $\sum_{\{\Gamma_1, \ldots, \Gamma_n \mid s_0 \stackrel{\Gamma_1}{\to} \cdots \stackrel{\Gamma_n}{\to} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}) PT(\Gamma_{n+1}, s_n) =$
  $\sum_{\{\Gamma_1, \ldots, \Gamma_n \mid s_0 \stackrel{\Gamma_1}{\to} \cdots \stackrel{\Gamma_n}{\to} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \Big[\prod_{i=1}^{n} PT(\Gamma_i, s_{i-1})$
  $\sum_{\{\Gamma_{n+1} \mid s_n \stackrel{\Gamma_{n+1}}{\to} s_{n+1},\ \mathcal{L}(\Gamma_{n+1}) = A_{n+1},\ s_n \in \mathcal{H}_n,\ s_{n+1} \in \mathcal{H}_{n+1}\}} PT(\Gamma_{n+1}, s_n)\Big] =$
  $\sum_{\{\Gamma_1, \ldots, \Gamma_n \mid s_0 \stackrel{\Gamma_1}{\to} \cdots \stackrel{\Gamma_n}{\to} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}) PM_{A_{n+1}}(s_n, \mathcal{H}_{n+1}) =$
  $\sum_{\{\Gamma_1, \ldots, \Gamma_n \mid s_0 \stackrel{\Gamma_1}{\to} \cdots \stackrel{\Gamma_n}{\to} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}) PM_{A_{n+1}}(\mathcal{H}_n, \mathcal{H}_{n+1}) =$
  $PM_{A_{n+1}}(\mathcal{H}_n, \mathcal{H}_{n+1}) \sum_{\{\Gamma_1, \ldots, \Gamma_n \mid s_0 \stackrel{\Gamma_1}{\to} \cdots \stackrel{\Gamma_n}{\to} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}) =$
  $PM_{A_{n+1}}(\mathcal{H}_n, \mathcal{H}_{n+1}) \prod_{i=1}^{n} PM_{A_i}(\mathcal{H}_{i-1}, \mathcal{H}_i) = \prod_{i=1}^{n+1} PM_{A_i}(\mathcal{H}_{i-1}, \mathcal{H}_i)$.

Let $s_0, \bar{s}_0 \in \mathcal{H}_0$. We have $PT(A_1 \cdots A_n, s_0) = \sum_{\{\Gamma_1, \ldots, \Gamma_n \mid s_0 \stackrel{\Gamma_1}{\to} \cdots \stackrel{\Gamma_n}{\to} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ (1 \le i \le n)\}} \prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}) =$
$\sum_{\mathcal{H}_1, \ldots, \mathcal{H}_n} \sum_{\{\Gamma_1, \ldots, \Gamma_n \mid s_0 \stackrel{\Gamma_1}{\to} \cdots \stackrel{\Gamma_n}{\to} s_n,\ \mathcal{L}(\Gamma_i) = A_i,\ s_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^{n} PT(\Gamma_i, s_{i-1}) =$
$\sum_{\mathcal{H}_1, \ldots, \mathcal{H}_n} \prod_{i=1}^{n} PM_{A_i}(\mathcal{H}_{i-1}, \mathcal{H}_i) =$
$\sum_{\mathcal{H}_1, \ldots, \mathcal{H}_n} \sum_{\{\overline{\Gamma}_1, \ldots, \overline{\Gamma}_n \mid \bar{s}_0 \stackrel{\overline{\Gamma}_1}{\to} \cdots \stackrel{\overline{\Gamma}_n}{\to} \bar{s}_n,\ \mathcal{L}(\overline{\Gamma}_i) = A_i,\ \bar{s}_i \in \mathcal{H}_i\ (1 \le i \le n)\}} \prod_{i=1}^{n} PT(\overline{\Gamma}_i, \bar{s}_{i-1}) =$
$\sum_{\{\overline{\Gamma}_1, \ldots, \overline{\Gamma}_n \mid \bar{s}_0 \stackrel{\overline{\Gamma}_1}{\to} \cdots \stackrel{\overline{\Gamma}_n}{\to} \bar{s}_n,\ \mathcal{L}(\overline{\Gamma}_i) = A_i,\ (1 \le i \le n)\}} \prod_{i=1}^{n} PT(\overline{\Gamma}_i, \bar{s}_{i-1}) = PT(A_1 \cdots A_n, \bar{s}_0)$.
Since we have the previous equality for all $s_0, \bar{s}_0 \in \mathcal{H}_0$, we can denote $PT(A_1 \cdots A_n, \mathcal{H}_0) = PT(A_1 \cdots A_n, s_0) = PT(A_1 \cdots A_n, \bar{s}_0)$.

By Proposition 7.1, $\sum_{s \in \mathcal{H} \cap DR(G)} \psi(s) = \sum_{s' \in \mathcal{H} \cap DR(G')} \psi'(s')$. Now we can complete the proof:
$\sum_{s \in \mathcal{H} \cap DR(G)} \psi(s) PT(\Sigma, s) = \sum_{s \in \mathcal{H} \cap DR(G)} \psi(s) PT(\Sigma, \mathcal{H}) = PT(\Sigma, \mathcal{H}) \sum_{s \in \mathcal{H} \cap DR(G)} \psi(s) =$
$PT(\Sigma, \mathcal{H}) \sum_{s' \in \mathcal{H} \cap DR(G')} \psi'(s') = \sum_{s' \in \mathcal{H} \cap DR(G')} \psi'(s') PT(\Sigma, \mathcal{H}) = \sum_{s' \in \mathcal{H} \cap DR(G')} \psi'(s') PT(\Sigma, s')$. ◻

Technical Reports

Fakultät II, Department für Informatik, Universität Oldenburg,
Postfach 2503, 26111 Oldenburg, Germany

1/87 A. Viereck: "Klassifikationen, Konzepte und Modelle für den Mensch-Rechner-Dialog" (Dissertation)

2/87 A. Schwill: "Forbidden subgraphs and reduction systems: A comparison"

3/87 J. Kämper: "Non-uniform proof systems: A new framework to describe non-uniform and probabilistic complexity classes"

1/88 K. Ambos-Spies, H. Fleischhack, H. Huwig: "Diagonalizing over deterministic polynomial time"

2/88 A. Schwill: " Shortest edge-disjoint paths in geodetically connected graphs"

3/88 V. Claus, U. Lichtblau (Hrsg.): "1. Tagung zur Küsten-Informatik"

1/89 U. van der Valk: "Einige Entscheidbarkeits- und Unentscheidbarkeitsresultate für Klasse von S/T-Netzen unter Maximum Firing Strategie und unter Prioritätenstrategien"

2/89 J. Kämper: "Strukturelle Untersuchungen im Umfeld der Komplexitätsklassen P und NP unter besonderer Berücksichtigung nichtuniformer, probabilistischer und disjunktiv selbstreduzierender Algorithmen" (Dissertation)

3/89 J. Kämper: "Nondeterministic oracle Turing machines with maximal computation paths"

1/90 A. Schwill: "Shortest edge-disjoint paths in graphs" (Dissertation)

2/90 K.R. Apt, E.-R. Olderog: "Using transformations to verify parallel programs"

3/90 U. Lichtblau: "Flußgraphgrammatiken" (Dissertation)

4/90 K.R. Apt, E.-R. Olderog: "Introduction to program verification"

5/90 H. Jasper: "Datenbankunterstützung für Prolog-Programmierumgebungen" (Dissertation)

1/91 F. Korf: "Net-based efficient simulation of AADL specifications"

2/91 S.V. Krishnan, C. Pandu Rangan, A. Schwill, S. Seshadri: "Two disjoint paths in chordal graphs"

3/91 H. Eirund: "Modellierung und Manipulation multimedialer Dokumente" (Dissertation)

4/91 G. Schreiber: "Ein funktionaler Äquivalenzbegriff für den hierarchischen Entwurf von Netzen"

1/92 A. Viereck (Hrsg.): "Ergebnisse der 11. Arbeitstagung, Mensch-Maschine Kommunikation"

2/92 P. Gorny, U. Daldrup, H. Schwab: "Zwischenbilanz: Menschengerechte Gestaltung von Software"

3/92 E.-R. Olderog, St. Rössig, J. Sander, M. Schenke: "ProCoS at Oldenburg: The Interface between Specification Language and occam-like Programming Language"

4/92 F. Korf: "Synthesis of VHDL Test Environments form Temporal Logic Specifications"

5/92 W. Kowalk: "Konstruktorentechnik: Neue Methoden zur Mengenrechnung, Logikrechnung und Intervallrechnung"

1/93 Ch. Dietz, G. Schreiber: "Eine Termdarstellung für S/T-Netze"

2/93 J. Sauer: "Wissensbasiertes Lösen von Ablaufplanungsproblemen durch explizite Heuristiken"

3/93 M. Sonnenschein, U. Lichtblau (Hrsg.): "6. Kolloquium der Arbeitsgruppe Informatik-Systeme"

4/93 H. Fleischhack, U. Lichtblau, M. Sonnenschein, R. Wieting: "Generische Definition hierarchischer zeitbeschrifteter höherer Petrinetze"

5/93 F. Köster, L. Twele, R. Wieting, W. Ziegler: "Fallbeispiele zur Modellierung mit THORNetzen"

1/94 R. Götze: "Dialogmodellierung für multimediale Benutzerschnittstellen"

2/94 B. Müller: "PPO-Eine objektorientierte Prolog-Erweiterung zur Entwicklung wissensbasierter Anwendungssysteme"

3/94 W. Damm/A. Mikschl: "Projekt Entwurf und Implementierung eines Multi-threaded RISC-Prozessors"

4/94 S. Rössig: "A Transformational Approach to the Design of Communicating Systems" (Dissertation)

5/94 G. Schreiber: "Funktionale Äquivalenz von Petri-Netzen" (Dissertation)

1/95 A. Gronewold, H. Fleischhack: "Language Preserving Reductions of Safe Petri-Nets"

2/95 H. Reineke: "Struktur und Verhalten von verteilten endlichen Automaten" (Dissertation)

3/95 H. Behrends: "Beschreibung ereignisgesteuerter Aktivitäten in datenbankgestützten Informationssystemen" (Dissertation)

4/95 U. M. Levens: "Computerunterstütztes Modellieren von Musikstücken mit Petri-Netzen: Das Mailänder Konzept"

1/96 M. Burke: "FDDI und ATM in multimedialen Anwendungsumgebungen" (Dissertation)

2/96 I. Pitschke: "Interaktive Rekonstruktion geometrischer Modelle aus digitalen Bildern" (Dissertation)

1/97 L. Bölke: "Ein akustischer Interaktionsraum für blinde Rechnerbenutzer" (Dissertation)

2/97 S. Schöf: "Verteilte Simulation höherer Petrinetze" (Dissertation)

1/98 S. Kleuker: "Inkrementelle Entwicklung von verifizierten Spezifikationen für verteilte Systeme" (Dissertation)

2/98 J. Bohn: “Mechanical Support and Validation of a Design Calculus for Communicating Systems by a Logic-Based Proof System“ (Dissertation)

3/98 L. Köhler: „Fuzzy Geometrie und Anwendungen in der medizinischen Bildverarbeitung“ (Dissertation)

4/98 J. Helbig: „Linking Visual Formalisms: A ‘Compositional Proof System for Statecharts Based on Symbolic Timing Diagrams“ (Dissertation)

5/98 G. Stiege: „Edge Partitions in Undirected Graphs“

6/98 A. Gerns: „Entwicklung und Bewertung von Objektmigrationsstrategien für verteilte Umgebungen“

7/98 M. Stadler: „Abstrakte Rechnernetzmodelle als Grundlage einer umfassenden Automatisierung des Netzmanagements – Konzepte und Sprachen zu ihrer Umsetzung“ (Dissertation)

8/98 M.-S. Steiner: „Lastverteilung in heterogenen Systemen“

9/98 Clemens Otte: „Fuzzy-Prototyp-Klassifikatoren und deren Anwendung zur automatischen Merkmalsselektion“

1/99 Juliane Vorndamme: „Die Auswirkungen rechtlicher Verpflichtungen auf die Software-entwicklung“

2/99 E. Best/K.M. Richter: „ Relational Semantics Revisited“

3/99 J. S. Lie: „Einsatz von Objektmigrationssystemen zur Leistungssteigerung in verteilten Systemen“

4/99 ZweiJahresbericht des Fachbereichs Informatik

5/99 Ingo Stierand, Olaf Maibaum, Björn Briel, Günther Stiege: „Cassandra - Generierung, Analyse und Simulation von eingebetteten Multiprozessor-Echtzeitsystemen“

6/99 Gunnar Wittich: „Ein problemorientierter Ansatz zum Nachweis von Realzeiteigenschaf-ten eingebetteter Systeme“

7/99 Annegret Habel, Jürgen Müller, Detlef Plump: „Double-Pushout Graph Transformation Revisited“

8/99 Ingo Stierand: „Eine Konfigurationssprache zur Erstellung von Ambrosia/MP-Systemen“

9/99 Igor V. Tarasyuk: „Equivalences for Concurrent and Distributed Systems“

10/99 Eike Best, Alexander Lavrov: „Generalised Composition Operations for High-Level Petri-Nets“

11/99 Alexander Lavrov: „Enhancing Mixed Nonlinear Optimization: A Hybrid Approach“

12/99 Alexander Lavrov: „Hybrid Techniques in Discrete-Event System Modelling and Control: some Examples“

13/99 Eike Best, Raymond Devillers, Maciej Koutny: „Recursion and Petri Nets“

14/99 Eike Best, Raymond Devillers, Maciej Koutny: „The Box Algebra = Petri Nets + Process Expressions“

15/99 Eike Best, Harro Wimmel: „Reducing *k*-safe Petri Nets to Pomset-equivalent 1-safe Petri Nets“

16/99 Udo Brockmeyer: „Verifikation von STATEMATE Designs“ (Dissertation)

1/00 Henning Dierks: „Specification and Verification of Polling Real-Time Systems“ (Dissertation)

2/00 Clemens Fischer: „Combination and Implementation of Processes and Data: from CSP-OZ to Java“ (Dissertation)

3/00 Cheryl Kleuker: „Constraint Diagrams“ (Dissertation)

4/00 Thomas Thielke: „Linear-algebraische Methoden zur Beschreibung, Verfeinerung und Analyse gefärbter Petrinetze“ (Dissertation)

1/01 Günther Stiege: „Higher Decomposition in Undirected Graphs“ (Bericht)

2/01 Ute Vogel: Zwei-Jahres-Bericht

3/01 Josef Tapken: „Model-Checking of Duration Calculus Specifications“ (Dissertation)

4/01 Björn Briel: „Analyse eingebetteter Systeme mittels verteilter Simulation“ (Dissertation)

5/01 Günther Stiege: „Standard Decomposition and Periodicity of Digraphs“ (Bericht)

6/01 Ingo Stierand: „Ambrosia/MP - Ein Echtzeitbetriebssystem für eingebettete Mehrprozessorsysteme„ (Dissertation)

1/02 Giorgio Busatto, Annegret Habel: „Improving the Quality of Hypertexts Using Graph Transformation“ (Bericht)

2/02 Giorgio Busatto: „Modeling Hyperweb Dynamics through Hierarchical Graph Transformation“ (Bericht)

3/02 Giorgio Busatto: „An Abstract Model of Hierarchical Graphs and Hierarchical Graph Transformation“ (Dissertation)

4/02 Laila Kabous: „An Object Oriented Design methodology for hard real Time Systems: The OOHARTS approach“ (Dissertation)

1/03 Ute Vogel: „2-Jahres-Bericht“

2/03 Olaf Maibaum: „Bestimmung symbolischer Laufzeiten in eingebetteten Echtzeitsystemen“ (Dissertation)

3/03 Günther Stiege, Ingo Stierand: „Connectedness-Based Hierarchical Decomposition of Undirected Graphs“ (Bericht)

4/03 Willi Hasselbring, Susanne Petersen: „Standards für die medizinische Kommunikation und Dokumentation (Bericht)

5/03 Andreas Möller: „Eine virtuelle Maschine für Graphprogramme“ (Bericht)

6/03 Tom Bienmüller: „Reducing Complexity for the Verification of Statemate Designs“ (Bericht)

7/03 Sandra Steinert: „Graph Programs for Graph Algorithms“ (Bericht)

8/03 Jochen Klose: „Live Sequence Charts: A Graphical Formalism for the Specification of Communication Behavior“ (Dissertation)

1/04 Jens Oehlerking: „Transformation of Edmonds’ Maximum Matching Algorithm into a Graph Program“ (Bericht)

2/04 Sergej Alekseev: „Dienste Intelligenter Netze Graphentheoretische Methoden in der Kontrollflussanalyse“ (Bericht)

3/04 Giorgio Busatto: „GraJ: A System für Executing Graph Programs in Java“ (Bericht)

1/05 Sergej Alekseev and Johannes Wust: „Graph Theoretical Methods in the Control Flow Analysis of Object Oriented Real Time Software“ (Bericht)

2/05 Ute Vogel: „2-Jahres-Bericht“

3/05 Igor Tarasyuk: „Discrete time stochastic Petri box calculus“ (Bericht)

1/06 Henning Dierks: „Time, Abstraction and Heuristics“ (Habilitation)

2/06 Li Sek Su: „Full-Output Siphons and Deadlock-Freeness for Free Choice Petri Nets“ (Bericht)

3/06 Timo Warns: „Solving Consensus Using Structural Failure Models„ (Bericht)

4/06 Sergej Alekseev: „Graphentheoretische Methoden in der Ablaufanalyse objektorientierter Anwendungen“ (Dissertation)

5/06 Li Sek Su: „Some Considerations on the Foundation of NP-Completeness Theory*“ (Bericht)

6/06 Li Sek Su: „Semitraps and Deadlock-Freeness for Reduced Asymmetric Choice Nets“ (Bericht)

7/06 Li Sek Su: „Algorithms of computing the Deadlock Markings Sets for Petri Nets“ (Bericht)

8/06 Annegret Habel, Karl-Heinz Pennemann, and Arend Rensink: „Weakest Preconditions for High-Level Programs (Long Version)“ (Bericht)

9/06 Jochen Hoenicke: „Combination of Processes, Data, and Time“ (Dissertation)

10/06 Steffen Becker, Marco Boscovic, Abhishek Dhama, Simon Giesecke, Jens Happe, Wilhelm Hasselbring, Heiko Koziolek, Henrik Lipskoch, Roland Meyer, Margarethe Muhle, Alexandra Paul, Jan Ploski, Matthias Rohr, Mani Swaminathan, Timo Warns, Daniel Winteler: „Trustworthy Software Systems: A Discussion of Basic Concepts and Terminology“ (Bericht)

11/06 Christian Zuckschwerdt: „Ein System zur Transformation von Konsistenz- in Anwendungs-bedingungen“ (Bericht)

01/07 Andreas Schäfer: „Specification and Verification of Mobile Real-Time Systems“ (Dissertation)

02/07 Günther Stiege: „General Graphs“ (Bericht)

03/07 Wolfgang Kowalk: „Integralrechnung“ (Bericht)

04/07 Karl Azab, Karl-Heinz Pennemann: „Type Checking C++ Template Instantiation by Graph Programs“ (Bericht)

01/08 Roland Meyer: „On depth and breath in the Pi-Calculus“ (Bericht)

02/08 Ingo Brückner: „Slicing Integrated Formal Specifications for Verification“ (Dissertation)

03/08 Ute Vogel: „2-Jahres-Bericht 2004 - 2006“ (Bericht)

04/08 Günther Stiege: „Summierbare Familien“ (Bericht)

05/08 Igor V. Tarasyuk: „Investigating equivalence relations in dtsPBC“ (Bericht)

01/09 Elke Wilkeit: „2-Jahres-Bericht 2007 – 2008“ (Bericht)

02/09 Roland Meyer: „Structural Stationarity in the pi-Calculus“ (Dissertation)

03/09 InformatikerInnen des Moduls Soft Skills: “E-Book Soft Skills 2008” (Bericht)

04/09 Eike Best: „Separability in Persistent Petri Nets“ (Bericht)

01/10 Igor Tarasyuk: „Equivalence relations for behaviour-preserving reduction and modular performance evaluation in dtsPBC, (Bericht)

02/10 Roman Dubtsov: „Timed Transition Systems with Independence and Marked Scott Domains: an Adjunction“ (Bericht)

01/11 Elena S. Oshevskaya: „Matching Equivalences on Higher Dimensional Automata Models“ (Bericht)

02/11 Elke Wilkeit: „2-Jahres-Bericht, 01.10.2008 - 30.09.2010“ (Bericht)

03/11 Johannes Faber: „Verification Architectures for Complex Real-Time Systems“ (Dissertation)