# Report on a research project supported by a DAAD scholarship

Dr. 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
http://www.iis.nsk.su/persons/itar/

## 1 Preliminaries

### 1.1 Scholarship specification

German Academic Exchange Service (Deutscher Akademischer Austauschdienst), grant A/08/08590, "Re-invitations for Former Scholarship Holders" ("Wiedereinladungen für ehemalige Stipendiaten").

### 1.2 Period of research

The work reported has been done within 2 months, between September 1 and October 31, 2008.

### 1.3 Supervisor

Prof. Dr. Eike Best, Parallel Systems Group, Theoretical Computer Science, Computer Science Department, Faculty II, Carl von Ossietzky University of Oldenburg (CvO UO), D-26111 Oldenburg, Germany, http://parsys.informatik.uni-oldenburg.de/persons/eike.best/.

## 2 The research area, motivation and previous work

Algebraic calculi hold a special place among formal models for specification of concurrent systems and analysis of their behavioral properties. In such process algebras (PAs), a system or a process is specified by an algebraic formula. A verification of the properties is accomplished at a syntactic level by means of well-developed systems of equivalences, axioms and inference rules. One of the first PAs was $CCS$ (Calculus of Communicating Systems) [9].

Process algebras have been acknowledged as very suitable formalism to operate with real time and stochastic systems as well. In the last years, stochastic extensions of PAs called stochastic process algebras (SPAs) became very popular as a modeling framework. SPAs do not just specify actions which can happen (qualitative features) as usual process algebras, but they associate some quantitative parameters with actions (quantitative characteristics). The most popular SPAs proposed so far are $PEPA$ [6], $TIPP$ [7] and $EMPA$ [2].

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

As a rule, stochastic process calculi proposed in the literature are based on interleaving. As a semantic domain, the interleaving formalism of transition systems is often used. Therefore, investigation of a stochastic extension for more expressive and powerful algebraic calculi is an important issue. At present, the development of step or "true concurrency" (such that parallelism is considered as a causal independence) SPAs is in the very beginning. At the same time, there were no algebra of infinite concurrent stochastic processes until past several years.

Petri box calculus ($PBC$) is a flexible and expressive process algebra based on calculus $CCS$. $PBC$ was proposed more than 15 years ago [1], and it was well explored since that time. It was intended to become a tool for description of a Petri net structure and relationships between nets. For $PBC$, denotational semantics in terms of a subclass of Petri nets equipped with interface and called Petri boxes was proposed. Calculus $PBC$ has step operational semantics in terms of labeled transition systems based on rules of structural operational semantics (SOS).

A notion of equivalence is very important in formal theory of computing processes and systems. Behavioural equivalences are applied during verification stage both to compare behaviour of systems and reduce their structure. At present time, there exists a wide diversity of equivalence notions for concurrent systems, and their interrelations were well explored in the literature. The most well-known and widely used one is bisimulation. Unfortunately, the mentioned behavioural equivalences take into account only functional (qualitative) but not performance (quantitative) aspects of system behaviour. Additionally, the equivalences are often interleaving ones, and they do not respect concurrency. In [3], a notion of interleaving stochastic bisimulation equivalence for process terms was introduced. At the same time, no appropriate equivalence notion was defined for concurrent SPAs until recently.

The scientific problem addressed in the project is the design of parallel systems taking into account both qualitative and quantitative features of computing. The main objective is the development of suitable formal models and methods respecting quantitative requirements of concurrent and distributed systems. This provides one with an ability to construct and validate realistic computing processes. The purpose could be achieved with a combined application of net and algebraic approaches to specification and analysis of stochastic concurrent systems. The basic models to be used are SPNs (labeled discrete time SPNs abbreviated as LDTSPNs) and SPAs (discrete time stochastic extension of $PBC$ $dtsPBC$) with step semantics.

We consider as very desirable to propose an equivalence relation for parallel SPAs that relates formulas specifying processes with similar behavior. The equivalences could be also used for behaviour-preserving reduction of stochastic processes. It is important to find a relation that is a congruence with respect to the algebraic operations. A characterization of equivalences via modal logics is used to change the operational reasoning on systems behaviour by the logical one that is a standard one for formal verification. On the other hand, we have an operational characterization of logical equivalences as a result. An investigation of stochastic processes in their steady states is a commonly used viewpoint for their performance evaluation via calculation of performance indices. It is very interesting to find which relations guarantee an identity of stationary behaviour of two equivalent processes for all their equivalence classes. Application examples and case studies are the essential parts of a theoretical investigation. They demonstrate how theoretical results can be used in practice.

In [11, 13] we presented a discrete time stochastic extension $dtsPBC$ of finite $PBC$. Step operational semantics was defined in terms of labeled transition systems based on action and inaction rules. Denotational semantics was defined in terms of a subclass of labeled DTSPNs (LDTSPNs) called discrete time stochastic Petri boxes (dts-boxes). In addition, we defined a variety of probabilistic equivalences that allow one to identify stochastic processes with similar behaviour that are differentiated by too strict notion of the semantic equivalence. The interrelations of all the introduced equivalences were investigated. In [12] an enrichment of $dtsPBC$ was constructed with the iteration operator used to specify infinite processes.

# 3 The results obtained

In this section we present a short overview of the results. The complete and self-contained technical report can be downloaded as `http://www.iis.nsk.su/persons/itar/dtspbcit_cov.pdf`.

1. Logical characterization of stochastic equivalences for $dtsPBC$.

   We have presented a characterization of algebraic probabilistic bisimulation equivalences of $dtsPBC$ via new probabilistic modal logics based on $PML$. The logic $iPML$ characterizes interleaving ($\underline{\leftrightarrow}_{is}$), and $PML$ characterizes step ($\underline{\leftrightarrow}_{ss}$) stochastic bisimulation equivalences.

   **Definition 3.1** *For a dynamic expression $G$ we write $G \models_G \Phi$, if $s_G \models_G \Phi$. Two dynamic expressions $G$ and $G'$ are* logically equivalent *in $iPML$, denoted by $G =_{iPML} G'$, if $\forall \Phi \in$ **iPML** $G \models_G \Phi \Leftrightarrow G' \models_{G'} \Phi$.*

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

   **Theorem 3.1** *For image-finite dynamic expressions $G$ and $G'$*

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

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

**Definition 3.2** *For a dynamic expression $G$ we write $G \models_G \Phi$, if $s_G \models_G \Phi$. Two dynamic expressions $G$ and $G'$ are* logically equivalent *in $sPML$, denoted by $G =_{sPML} G'$, if $\forall \Phi \in$ **sPML** $G \models_G \Phi \Leftrightarrow G' \models_{G'} \Phi$.*

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

**Theorem 3.2** *For image-finite dynamic expressions $G$ and $G'$*

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

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

2. Steady states and application of the stochastic equivalences to comparing stationary behaviour.

   We have proved that step stochastic bisimulation equivalence guarantees similarity of stationary behaviour on the equivalence classes as composite states. In particular, for two processes related by step stochastic bisimulation equivalence the overall steady state probabilities to come in an equivalence class coincide. Further, it has been demonstrated that for the mentioned processes the steady state probabilities to come in an equivalence class and start a step trace from it are equal.

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

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

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 step trace from it coincide.

**Theorem 3.3** *Let $G, G'$ be dynamic expressions with $\mathcal{R} : G \underline{\leftrightarrow}_{ss} G'$ and $\Sigma$ be a step trace. Then $\forall \mathcal{H} \in (DR(G) \cup DR(G'))/_{\mathcal{R}}$*

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

3. Preservation of the equivalences by algebraic operations and constructing the congruence relation.

   We have investigated which equivalences of *dtsPBC* withstand application of all the algebraic operations. Using this knowledge, we have constructed a new congruence relation for the calculus based on the extended notion of the transition system $TS(G)$ of a dynamic expression $G$ called *sr*-transition system and denoted by $TS_{sr}(G)$.

**Definition 3.3** *Let $E$ be a static expression and $TS(\overline{E}) = (S, L, \mathcal{T}, s)$. The* (labeled probabilistic) *sr*-transition system *of $\overline{E}$ is a quadruple $TS_{sr}(\overline{E}) = (S_{sr}, L_{sr}, \mathcal{T}_{sr}, s_{sr})$, where*

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

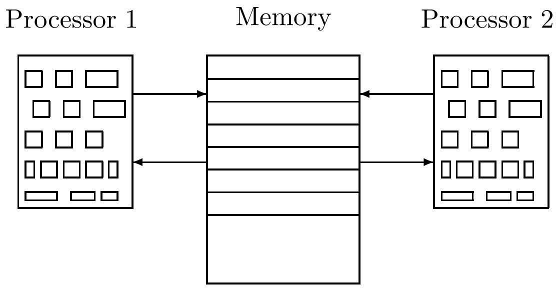


Figure 1: The diagram of the shared memory system

**Definition 3.4** *Two dynamic expressions* $\overline{E}$ *and* $\overline{E'}$ *are* isomorphic with respect to $sr$-transition systems, *denoted by* $\overline{E} =_{tssr} \overline{E'}$, *if* $TS_{sr}(\overline{E}) \simeq TS_{sr}(\overline{E'})$.

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

**Theorem 3.4** *Let* $a \in Act$ *and* $E, E', F, K \in RegStatExpr$. *If* $\overline{E} =_{tssr} \overline{E'}$ *then*

*(a)* $\overline{E \circ F} =_{tssr} \overline{E' \circ F}$, $\overline{F \circ E} =_{tssr} \overline{F \circ E'}$, $\circ \in \{;, [], \|\}$;

*(b)* $\overline{E[f]} =_{tssr} \overline{E'[f]}$;

*(c)* $\overline{E \circ a} =_{tssr} \overline{E' \circ a}$, $\circ \in \{\mathsf{rs}, \mathsf{sy}\}$;

*(d)* $\overline{[E * F * K]} =_{tssr} \overline{[E' * F * K]}$, $\overline{[F * E * K]} =_{tssr} \overline{[F * E' * K]}$, $\overline{[F * K * E]} =_{tssr} \overline{[F * K * E']}$.

4. Examples of specification, analysis and reduction, case studies: processes with shared memory and dining philosophers.

We have proposed two application examples based on process specifications of $dtsPBC$ that explain how to analyze performance of systems and their reductions w.r.t. step stochastic bisimulation equivalence within the calculus. We have considered algebraic models of shared memory system and dining philosophers one.

**Shared memory system**

Consider a model of two processors accessing a common shared memory described in [8] in the continuous time setting on GSPNs. We analyze this shared memory system in the discrete time setting within $dtsPBC$ where concurrent execution of activities is possible. The model performs as follows. After activation of the system (turning the computer on), two processors are active, and the common memory is available. Each processor can request an access to the memory. When a processor starts an acquisition of the memory, another processor should wait until the former one ends its memory operations, and the system returns to the state with both active processors and the available common memory. The diagram of the system is depicted in Figure 1.

The action $a$ corresponds to the system activation. The actions $r_i$ ($1 \leq i \leq 2$) represent the common memory request of processor $i$. The actions $b_i$ and $e_i$ correspond to the beginning and the end, respectively, of the common memory access of processor $i$. The other actions are used for communication purpose only via synchronization, and we abstract from them later using restriction.

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

In Figure 2 the transition system without empty loops $TS^*(\overline{E})$ is presented.

In Figure 3 the marked dts-box corresponding to the dynamic expression of the shared memory system is presented, i.e., $N = Box_{dts}(\overline{E})$.

**Dining philosophers system**

Consider a model of five dining philosophers, for which the Petri net interpretation was proposed in [10], in the discrete time stochastic setting of $dtsPBC$. The philosophers occupy a round table, and there is one

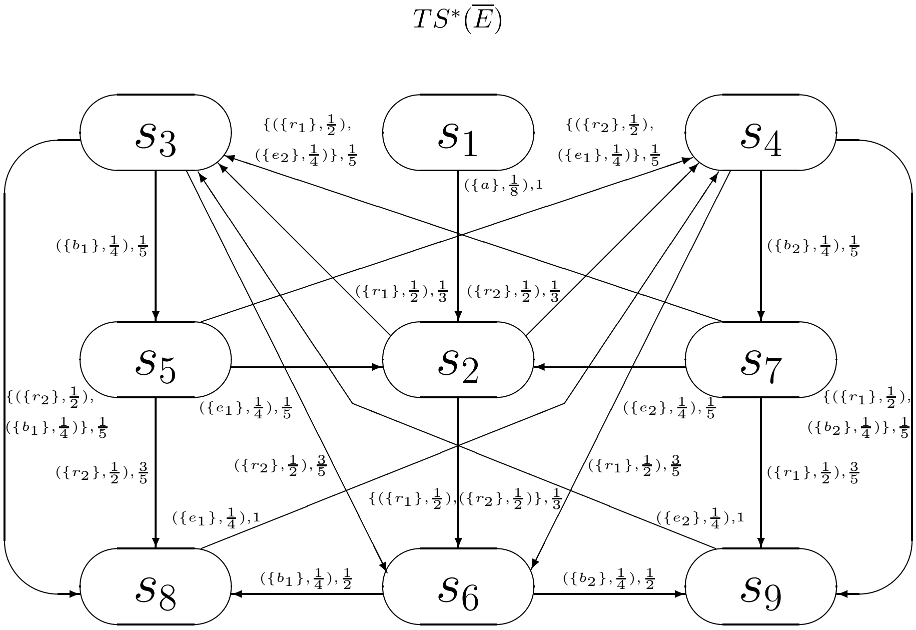


Figure 2: The transition system without empty loops of the shared memory system

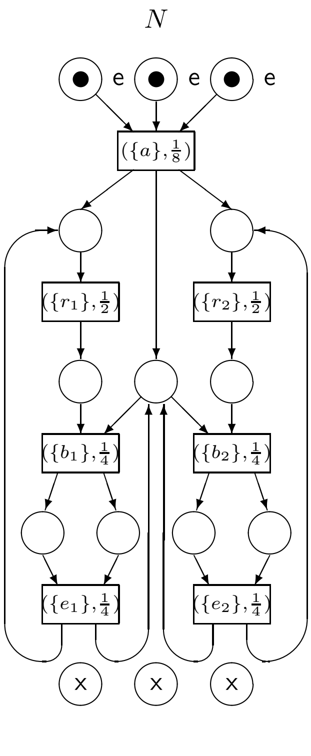


Figure 3: The marked dts-box of the shared memory system

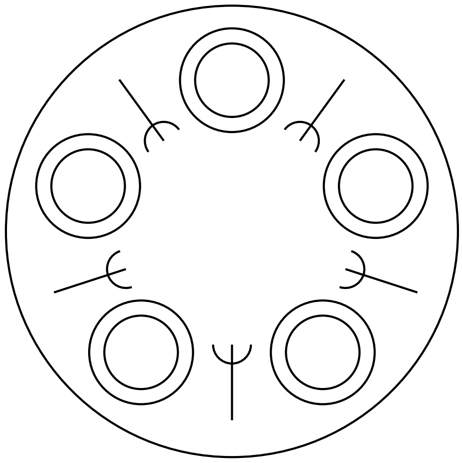

Figure 4: The diagram of the dining philosophers system

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 one. Hence, all five philosophers cannot eat together, since otherwise there will be not enough forks available, but only one of two of them who are not neighbors. The model performs as follows. After activation of the system (coming the philosophers in the dining room), five forks appear on the table. If the left and right forks available for a philosopher, he takes them simultaneously and begins eating. At the end of eating, the philosopher places both his forks simultaneously back on the table. The strategy to pick up and release two forks simultaneously prevents the situation when a philosopher takes one fork but is not able to pick up the second one since their neighbor has already done so. In particular, we avoid a deadlock when all the philosophers take their left (right) forks and wait until their right (left) forks will be available. The diagram of the system is depicted in Figure 4.

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 purpose 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 $E = (E_1 \| E_2 \| E_3 \| E_4 \| E_5)$ $\mathsf{sy}\ x_1$ $\mathsf{sy}\ x_2$ $\mathsf{sy}\ x_3$ $\mathsf{sy}\ x_4$ $\mathsf{sy}\ y_1$ $\mathsf{sy}\ y_2$ $\mathsf{sy}\ y_3$ $\mathsf{sy}\ y_4$ $\mathsf{sy}\ y_5$ $\mathsf{sy}\ z_1$ $\mathsf{sy}\ z_2$ $\mathsf{sy}\ z_3$ $\mathsf{sy}\ z_4$ $\mathsf{sy}\ z_5$ $\mathsf{rs}\ x_1$ $\mathsf{rs}\ x_2$ $\mathsf{rs}\ x_3$ $\mathsf{rs}\ x_4$ $\mathsf{rs}\ y_1$ $\mathsf{rs}\ y_2$ $\mathsf{rs}\ y_3$ $\mathsf{rs}\ y_4$ $\mathsf{rs}\ y_5$ $\mathsf{rs}\ z_1$ $\mathsf{rs}\ z_2$ $\mathsf{rs}\ z_3$ $\mathsf{rs}\ z_4$ $\mathsf{rs}\ z_5$.

In Figure 5 the transition system without empty loops $TS^*(\overline{E})$ is presented.

In Figure 6 the marked dts-box corresponding to the dynamic expression of the dining philosophers system is presented, i.e., $N = Box_{dts}(\overline{E})$.

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

The static expression of the philosopher $i$ ($1 \leq i \leq 4$) 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 $F = (F_1 \| F_2 \| F_3 \| F_4 \| F_5)$ $\mathsf{sy}\ x_1$ $\mathsf{sy}\ x_2$ $\mathsf{sy}\ x_3$ $\mathsf{sy}\ x_4$ $\mathsf{sy}\ y_1$ $\mathsf{sy}\ y_2$ $\mathsf{sy}\ y_3$ $\mathsf{sy}\ y_4$ $\mathsf{sy}\ y_5$ $\mathsf{sy}\ z_1$ $\mathsf{sy}\ z_2$ $\mathsf{sy}\ z_3$ $\mathsf{sy}\ z_4$ $\mathsf{sy}\ z_5$ $\mathsf{rs}\ x_1$ $\mathsf{rs}\ x_2$ $\mathsf{rs}\ x_3$ $\mathsf{rs}\ x_4$ $\mathsf{rs}\ y_1$ $\mathsf{rs}\ y_2$ $\mathsf{rs}\ y_3$ $\mathsf{rs}\ y_4$ $\mathsf{rs}\ y_5$ $\mathsf{rs}\ z_1$ $\mathsf{rs}\ z_2$ $\mathsf{rs}\ z_3$ $\mathsf{rs}\ z_4$ $\mathsf{rs}\ z_5$.

Let us consider a reduction of the abstract dining philosophers system.

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

We have $\overline{F} \underline{\leftrightarrow}_{ss} \overline{F'}$ with $(DR(\overline{F}) \cup DR(\overline{F'}))/_{\underline{\leftrightarrow}_{ss}} = \{\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4\}$, where $\mathcal{H}_1 = \{s_1, s_1'\}$ (the initial state), $\mathcal{H}_2 = \{s_2, s_2'\}$ (the system is activated and no philosophers dine), $\mathcal{H}_3 = \{s_3, s_6, s_7, s_{10}, s_{11}, s_3', s_4'\}$

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

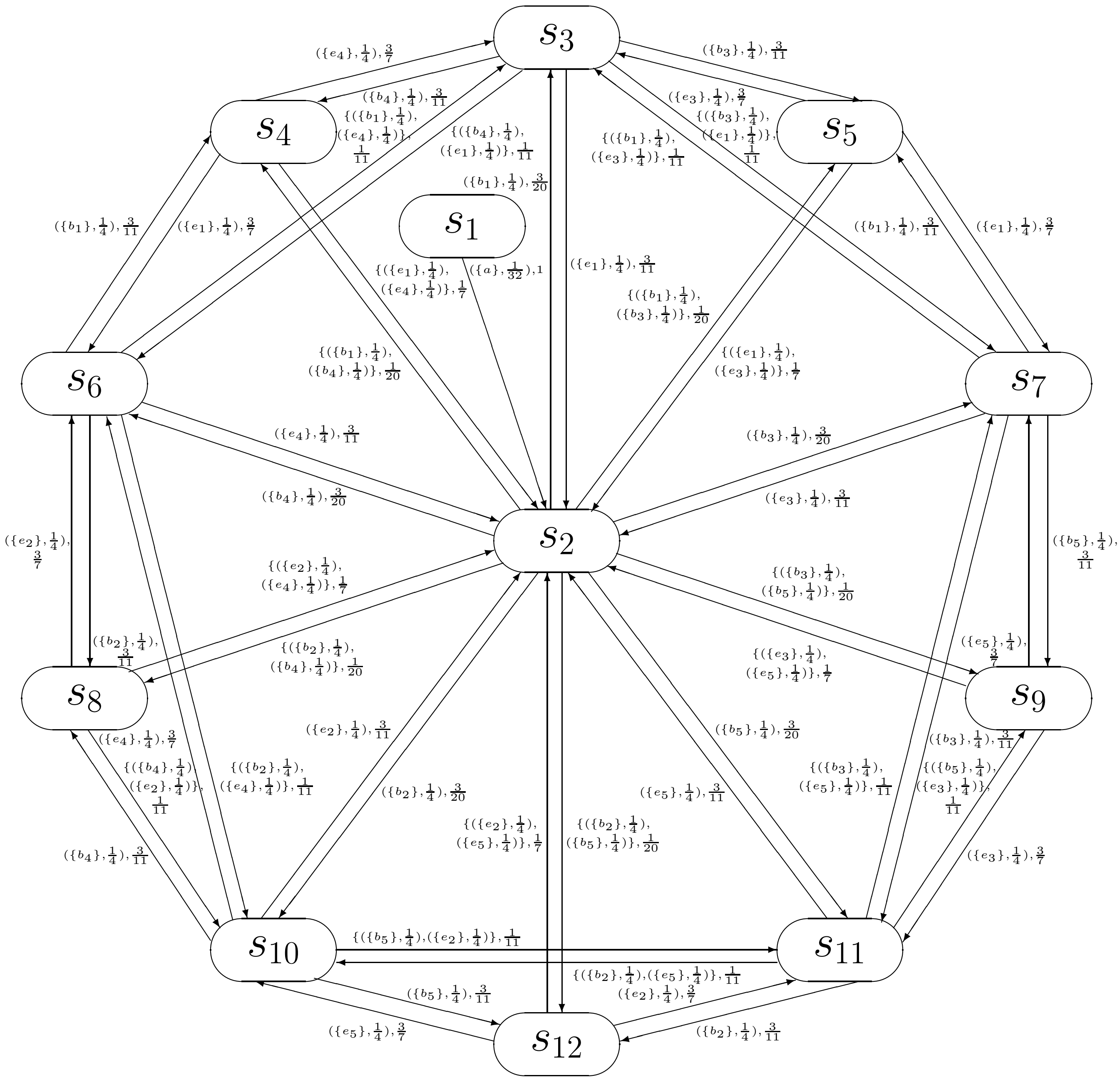


Figure 5: The transition system without empty loops of the dining philosophers system

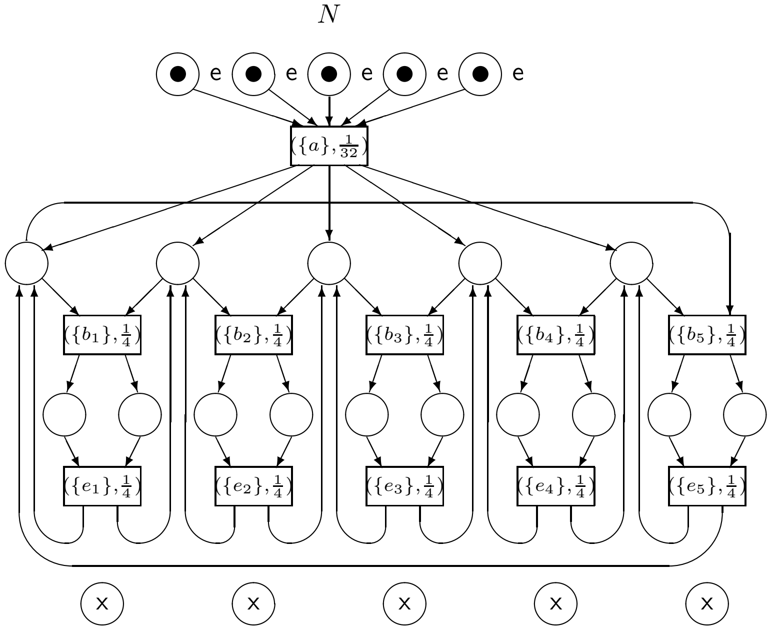


Figure 6: The marked dts-box of the dining philosophers system

(one philosopher dines), $\mathcal{H}_4 = \{s_4, s_5, s_8, s_9, s_{12}, s'_5\}$ (two philosophers dine). One can see that $F'$ is a reduction of $F$ with respect to $\underline{\leftrightarrow}_{ss}$.

In Figure 7 the transition system without empty loops $TS^*(\overline{F'})$ is presented.

In Figure 8 the marked dts-box corresponding to the dynamic expression of the reduced abstract dining philosophers system is presented, i.e., $N' = Box_{dts}(\overline{F'})$.

Note that $TS^*(\overline{F'})$ can be reduced further by merging the equivalent states $s'_3$ and $s'_4$, thus, it can be transformed into a transition system with four states only. But the resulted "minimal" reduction with respect to $\underline{\leftrightarrow}_{ss}$ of the initial transition system $TS^*(\overline{F})$ will not be anymore a transition system without empty loops corresponding to some $dtsPBC$ expression. Hence, in the general case, the procedure of expressions reduction cannot be transferred smoothly from a transition systems level. The minimal equivalent expression does not always have the minimal transition system, in the case the latter can be further reduced. In the following definition we consider step stochastic bisimulation equivalence between states of a dynamic expression.

**Definition 3.5** *The* minimal reduced with respect to $\underline{\leftrightarrow}_{ss}$ (labeled probabilistic) transition system without empty loops *of a dynamic expression $G$ 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*

- $S_{\underline{\leftrightarrow}_{ss}} = DR(G)/_{\underline{\leftrightarrow}_{ss}}$;
- $L_{\underline{\leftrightarrow}_{ss}} \subseteq I\!N_f^{\mathcal{L}} \times (0; 1]$;
- $\mathcal{T}_{\underline{\leftrightarrow}_{ss}} = \{(\mathcal{H}, (A, \mathcal{P}), \widetilde{\mathcal{H}}) \mid \exists s \in \mathcal{H}\ s \xrightarrow{A}_{\mathcal{P}} \mathcal{H}\}$;
- $s_{\underline{\leftrightarrow}_{ss}} = \{[G]_\simeq\}$.

We have $DR(\overline{F})/_{\underline{\leftrightarrow}_{ss}} = \{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_4\}$, 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 9 the minimal reduced with respect to $\underline{\leftrightarrow}_{ss}$ transition system without empty loops $TS^*_{\underline{\leftrightarrow}_{ss}}(\overline{F})$ is presented.

Obviously, it is easier to evaluate performance with the use of a DTMC with less states, since in this case the dimension of the transition probability matrix will be smaller. Hence, to calculate steady-state probabilities, we shall solve systems of less equations. Thus, we have obtained the following method of performance analysis simplification. First, we construct the minimal reduced with respect to $\underline{\leftrightarrow}_{ss}$

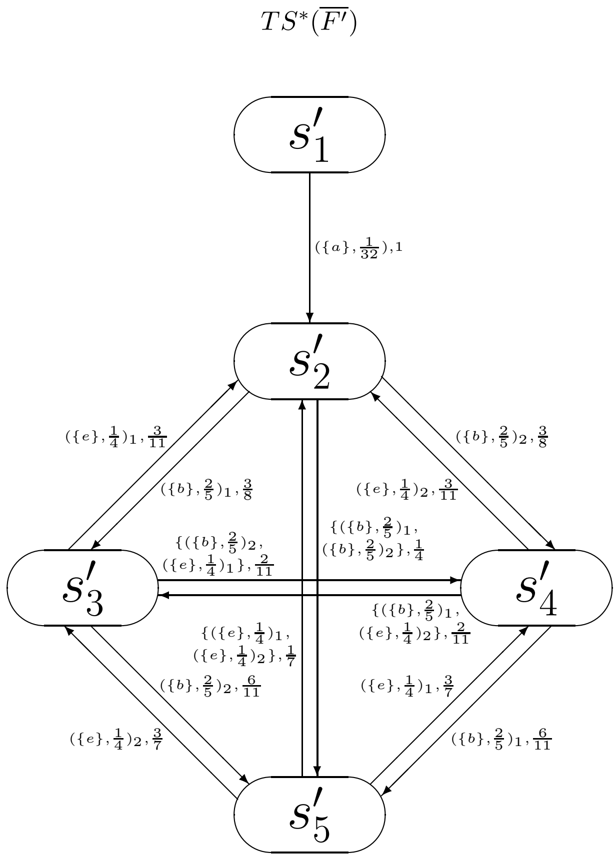


Figure 7: The transition system without empty loops of the reduced abstract dining philosophers system

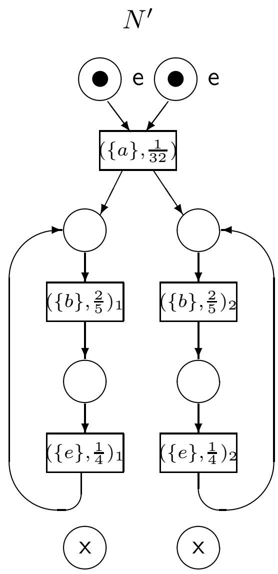


Figure 8: The marked dts-box of the reduced abstract dining philosophers system

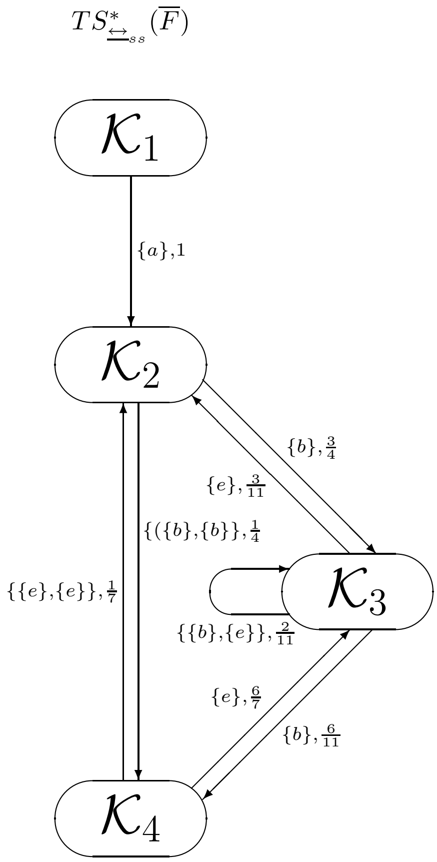


Figure 9: The minimal reduced with respect to $\underline{\leftrightarrow}_{ss}$ transition system without empty loops of the abstract dining philosophers system

underlying DTMC without empty loops. Second, we calculate steady-state probabilities and performance indices based on this minimal reduction DTMC. The indices will be the same as those calculated based on the initial unreduced DTMC.

# 4 The papers prepared

1. 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://www.iis.nsk.su/persons/itar/dtspbcit_cov.pdf`.

2. Tarasyuk I.V. *A notion of congruence for dtsPBC. Bulletin of the Novosibirsk Computing Center, Series Computer Science, IIS Special Issue* **28**, 20 pages, NCC Publisher, Novosibirsk, 2008 (submitted).

3. Tarasyuk I.V. *Modeling and analysis of computing systems in the algebra dtsPBC.* 19 pages, 2008 (in Russian, submitted).

# 5 Presentation of results

- Giving the talk "Performance evaluation in *dtsPBC*" (`http://www.iis.nsk.su/persons/itar/dtspbcsm_pe.pdf`), *Weekly Research Seminar of Graduate and Postgraduate Students ("Diplomanden- und Doktorandenseminar (D+D)"*, Computer Science Departmen, Faculty II, CvO UO, Oldenburg, Germany.

# 6 Participating regular research meetings

- *Weekly Internal Seminar of Parallel Systems Group*, Computer Science Department, Faculty II, CvO UO, Oldenburg, Germany.

- *Weekly Research Seminar of Graduate and Postgraduate Students ("Diplomanden- und Doktorandenseminar (D+D)"*, Computer Science Department, Faculty II, CvO UO, Oldenburg, Germany.

# 7 Future research

Future work consists in abstracting from the silent activities in the definitions of the equivalences, i.e., from the activities with empty multiaction part. The abstraction from empty loops and that from silent activities could be done in one step as well. The main point here is that we should collect probabilities during such the abstractions from an internal activity. As a result, we shall have the algebraic analogues of the net stochastic equivalences from [4, 5]. Moreover, we plan to extend *dtsPBC* with recursion to enhance specification power of the calculus. The research work mentioned above will be hopefully proceeded in a close cooperation with members of Parallel Systems Group, Computer Science Department, CvO UO, Oldenburg, Germany.

# References


[1] Best E., Devillers R., Hall J.G. *The box calculus: a new causal algebra with multi-label communication. Lecture Notes in Computer Science* **609**, p. 21–69, 1992.

[2] Bernardo M., Gorrieri R. *A tutorial on EMPA: a theory of concurrent processes with nondeterminism, priorities, probabilities and time. Theoretical Computer Science* **202**, p. 1–54, July 1998.

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

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

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

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

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

[8] Marsan M.A., Balbo G., Conte G., Donatelli S., Franceschinis G. *Modelling with generalized stochastic Petri nets. Wiley Series in Parallel Computing*, John Wiley and Sons, 316 p., 1995, `http://www.di.unito.it/~greatspn/GSPN-Wiley/`.

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

[10] Peterson J.L. *Petri net theory and modeling of systems.* Prentice-Hall, 1981.

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

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

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