

# Equivalence and Lumpability of FSPNs

Falko Bause<sup>1</sup>, Peter Buchholz<sup>1(✉)</sup>, Igor V. Tarasyuk<sup>2</sup>, and Miklós Telek<sup>3</sup>

<sup>1</sup> Informatik 4, TU Dortmund, 44221 Dortmund, Germany

`{falko.bause,peter.buchholz}@cs.tu-dortmund.de`

<sup>2</sup> A.P. Ershov Institute of Informatics Systems,

Siberian Branch of the Russian Academy of Sciences,

Acad. Lavrentiev pr. 6, 630090 Novosibirsk, Russian Federation

`itar@iis.nsk.su`

<sup>3</sup> MTA-BME Information Systems Research Group,

Department of Networked Systems and Services,

Technical University of Budapest, 1521 Budapest, Hungary

`telek@hit.hme.hu`

**Abstract.** We consider equivalence relations for Fluid Stochastic Petri Nets (FSPNs). Based on equivalence relations for Stochastic Petri Nets (SPNs), which are derived from lumpability for Markov Chains, and from lumpability for certain classes of differential equations, we define an equivalence relation for FSPNs. Lumpability for the differential equations is based on a finite discretization approach and permutations of the fluid part of the FSPN.

As for other modeling formalisms, the availability of an appropriate equivalence relation allows one to aggregate sets of equivalent states into single states. This state space reduction can be exploited for a more efficient analysis of FSPNs using a discretization approach. Lumpable equivalence relations can be computed from an appropriately discretized state space of the stochastic process or directly from the FSPN.

**Keywords:** Fluid Stochastic Petri Nets · Lumpability · Equivalence

## 1 Introduction

The idea of lumping states in a discrete system has a long history in Markov chains [1, 15] but has also been used in linear systems [6] and for differential equations [16]. Later it has been applied to specific modeling formalisms like stochastic process algebras [11] and even stochastic Petri nets [2]. Current developments can be found for fluid models [13, 17]. The central idea of lumpability is the definition of classes of states with an identical behavior and the substitution of the state classes by single states without altering the behavior of the system as it is observed. We present this approach for FSPNs in two versions. First we introduce a discretized version of the system and discuss lumping on the

---

The work of F. Bause, P. Buchholz and I. Tarasyuk has been partially supported by DFG under grant BE 1267/14-1.

discretized model. Next we discuss lumping on the original FSPN. The first approach involves identical behavior on the level ODEs while the latter one presents identical behavior on the level of PDEs.

*New Contribution of the Paper:* Lumpability has been applied in the above mentioned papers for discrete models or for specific types of continuous models as they result from kinetic differential equations [16]. The latter approach has then been used as a basis to define lumpability for a fluid description of stochastic process algebra terms which result from a large number of identical and symmetric components [13, 17]. Our approach combines lumpability for discrete and for continuous systems and presents, to the best of our knowledge for the first time, an approach that can be applied to hybrid systems. Additionally, the lumping of the continuous part goes beyond the approach presented for stochastic process algebras because lumping does not necessarily imply symmetry in the model.

The rest of the paper is organized as follows. FSPNs and the notation are introduced in Sect. 2. The analytical description of FSPNs by means of PDEs and a proposed discretizations approach resulting in an ODE-based analytical description are provided in Sect. 3. Lumpability of the discretized system is analyzed in Sect. 4, while in Sect. 5, lumpability is analyzed directly on the system matrices without the discretization step. The paper is concluded in Sect. 6.

## 2 Background and Definitions

We consider a class of FSPNs, which is similar to FSPNs, presented in [10, 12]. A FSPN is an 7-tuple  $(P, T, \bar{m}_0, A, B, F, R)$ , where

- $P$  is the set of places which is subdivided into the set of discrete places  $P_d$  and the set of continuous places  $P_c$ ,
- $T$  is the set of (timed) transitions,
- $\bar{m}_0 = (\mathbf{m}_0, \mathbf{x}_0)$  is the initial marking, where  $\mathbf{m}_0 \in \mathbb{N}^{|P_d|}$  is a vector containing the number of tokens on each discrete place and  $\mathbf{x}_0 \in \mathbb{R}^{|P_c|}$  is a vector which contains for each continuous place the level of fluid at the place. Let  $\mathcal{M}_d$  be the set of all reachable discrete markings,  $\mathcal{M}_c$  be the set of reachable continuous markings and  $\mathcal{M}$  the set of all markings,
- $A$  is the set of arcs which is subdivided into discrete arcs  $A_d : ((P_d \times T) \cup (T \times P_d)) \rightarrow \mathbb{N}$  (where  $A_d$  defines the multiplicity of the arc) and continuous arcs  $A_c : (P_c \times T) \cup (T \times P_c) \rightarrow \{0, 1\}$ ,
- $B$  is the set of capacities of fluid places, i.e.,  $B : P_c \rightarrow \mathbb{R}_{>0}$ ,
- $F$  the set of transition rates which is a function  $F : T \times \mathcal{M} \rightarrow \mathbb{R}_{\geq 0}$ ,
- $R$  the set of flow rates which is a function  $R : A_c \times \mathcal{M} \rightarrow \mathbb{R}_{\geq 0}$ .

We do not consider immediate transitions here which are commonly available in FSPNs (e.g., [10, 12]), because it is easier to define equivalence relations for FSPNs with only timed transitions. However, it is possible to extend the approach to FSPNs with immediate transitions. The marking dependent fluid rate is a very powerful concept. It allows one to model for example inhibitor arcs or place capacities for discrete places, both are not explicitly part of our class of nets.

For a transition  $t \in T$ , we denote the input places by  $\bullet t = \{p \in P_d | A_d(p, t) > 0\}$ , similarly the output places by  $t\bullet = \{p \in P_d | A_d(t, p) > 0\}$ . For continuous places the notation  $ot = \{p \in P_c | A_c(p, t) = 1\}$  and  $t\circ = \{p \in P_c | A_c(t, p) = 1\}$  are applied. The input and output transitions,  $\bullet p, p\bullet, op$  and  $p\circ$ , are defined similarly.

A transition  $t \in T$  is enabled in marking  $\bar{m} = (\mathbf{m}, \mathbf{x})$ , if for all  $p \in \bullet t$ ,  $A_d(p, t) \leq \mathbf{m}(p)$  and  $F(t, \bar{m}) > 0$ . Let  $ena(\bar{m})$  be the set of transitions enabled in marking  $\bar{m}$ . Enabled transitions may modify the discrete and continuous state (i.e., marking) of the net.

We start with the discrete part of the marking. The discrete part is modified by firing an enabled transition. Firing times are exponentially distributed with rate  $F(t, \bar{m})$  for  $t \in ena(\bar{m})$ . The transition that fires is selected according to a race condition. Firing transition  $t$  in marking  $\bar{m} = (\mathbf{m}, \mathbf{x})$  results in the new marking  $\bar{m}' = (\mathbf{m}', \mathbf{x})$  with  $\mathbf{m}'(p) = \mathbf{m}(p) - A_d(p, t) + A_d(t, p)$  for all  $p \in P_d$ . The enabling conditions of transitions assure that all components of  $\mathbf{m}'$  are non-negative. We use the notation  $\bar{m} \xrightarrow{t} \bar{m}'$  if  $t$  fires in  $\bar{m}$  and results in marking  $\bar{m}'$ . If only the discrete part is relevant we use the notation  $\mathbf{m} \xrightarrow{t} \mathbf{m}'$ . Observe that the firing of transitions does not modify the continuous state.

The continuous marking,  $\mathbf{x}$ , is continuously modified with a finite rate by enabled transitions, as long as the place capacities are respected. In marking  $\bar{m}$ , the potential flow rate for place  $p \in P_c$  is given by

$$\check{r}_p(\bar{m}) = \sum_{t \in ena(\bar{m}) \cap op} R((t, p), \bar{m}) - \sum_{t \in ena(\bar{m}) \cap po} R((p, t), \bar{m}). \quad (1)$$

The actual flow rate has to take care of the place capacities and is defined as

$$r_p(\bar{m}) = \begin{cases} \check{r}_p(\bar{m}) & \text{if } 0 < x_p < B(p), \\ \check{r}_p(\bar{m}) & \text{if } x_p = 0 \wedge \check{r}_p(\bar{m}) > 0, \\ \check{r}_p(\bar{m}) & \text{if } x_p = B(p) \wedge \check{r}_p(\bar{m}) < 0, \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where  $x_p$  is the fluid level at fluid place  $p$ . The rate describes the flow rate into a continuous place, i.e.,  $r_p(\bar{m}) = \frac{dx_p(\tau)}{d\tau}$ , where  $\tau$  denotes the time, and negative flow rate represents a decaying fluid level.

The model allows one to define some nasty behaviors, which means that flows or rates change infinitely often in a finite interval, as for example shown in [5]. We will exclude these behaviors in the following section and assume that the majority of the transition rate and flow rate functions are either independent or a piecewise constant function of the continuous marking  $\mathbf{x}$ . In principle, simulation can be applied to analyze FSPNs. We consider here numerical analysis via discretization where lumping helps to reduce the analysis complexity. To describe our approach we introduce several restrictions for the allowed class of nets. Some of these restrictions may be relaxed and still allow one to compute equivalence relations and analyze the resulting systems numerically, others are essential in the sense that otherwise a numerical analysis is no longer possible and an equivalence relation to reduce the state space can no longer be computed. The approach will be presented in the subsequent sections.

### 3 Discretization and Analysis

We consider only FSPNs with a finite set  $\mathcal{M}_d$  otherwise numerical analysis can only be applied in very specific cases. Generation of the set  $\mathcal{M}_d$  is in general non-trivial due to the presence of marking dependent transition rates which may become zero. However, it is easy to compute a super-set of  $\mathcal{M}_d$  by neglecting all continuous components in the net which means that enabling conditions of transitions that depend on the filling of fluid places are simply ignored. We assume in the sequel that  $\mathcal{M}_d$  or an appropriate finite super-set of  $\mathcal{M}_d$  can be generated using common algorithms for state space generation.

For  $\mathbf{x} \in \mathcal{M}_c$ ,  $\mathbf{Q}(\mathbf{x})$  is a  $|\mathcal{M}_d| \times |\mathcal{M}_d|$  matrix including the transitions rates if the continuous marking is  $\mathbf{x}$ . Markings from  $\mathcal{M}_d$  are numbered consecutively from 1 through  $|\mathcal{M}_d|$ . We use the marking  $\mathbf{m}_i$  and its number  $i$  interchangeably and have for the elements of matrix  $\mathbf{Q}(\mathbf{x})$

$$\begin{aligned} q_{ij}(\mathbf{x}) &= \sum_{t \in \text{ena}(\mathbf{m}_i) \wedge \mathbf{m}_i \xrightarrow{t} \mathbf{m}_j} F(t, (\mathbf{m}_i, \mathbf{x})), \text{ for } i \neq j, \\ q_{ii}(\mathbf{x}) &= - \sum_{j \neq i} q_{ij}(\mathbf{x}). \end{aligned} \quad (3)$$

Similarly, we define for each continuous place  $p \in P_c$  a diagonal matrix  $\mathbf{R}_p(\mathbf{x})$  of size  $|\mathcal{M}_d| \times |\mathcal{M}_d|$  with  $r_p((\mathbf{m}_i, \mathbf{x}))$  in position  $(i, i)$ .

FSPNs as we defined them allow for a very complex behavior where the flow rate and also transition rates depend on the filling of fluid places in an arbitrary complex way. In full generality, such a behavior can hardly be analyzed. Therefore we assume that  $\mathcal{M}_c$  can be decomposed in finitely many disjoint subsets  $\mathcal{M}_c^1, \dots, \mathcal{M}_c^K$  such that for  $\mathbf{x}, \mathbf{y} \in \mathcal{M}_c^k$   $\mathbf{Q}(\mathbf{x}) = \mathbf{Q}(\mathbf{y})$  and  $\mathbf{R}_p(\mathbf{x}) = \mathbf{R}_p(\mathbf{y})$  for all  $p \in P_c$  and  $\mathbf{m} \in \mathcal{M}_d$ . We assume that each subset  $\mathcal{M}_c^k$  is built from finite intervals  $(b_p^{k-1}, b_p^k)$  with  $b_p^{k-1} < b_p^k$  ( $1 \leq k < K$ ),  $b_p^0 = 0$ ,  $b_p^K = B(p)$  for  $p \in P_c$ . We note that probability mass of various dimensions, characterized by appropriate boundary equations, can develop at set boundaries, if some components of  $\mathbf{R}_p(\mathbf{x})$  changes sign, but their discussion we also neglect here. Thus, we assume that at  $b_p^k$  the functions in the matrices  $\mathbf{Q}(\mathbf{x})$  and  $\mathbf{R}_p(\mathbf{x})$  are left or right continuous or appropriate boundary conditions can be defined.

The dynamic behavior of FSPNs with more than one continuous place specifies a set of partial differential equations. The derivation for these equations will be briefly summarized and follows [3, 8–10, 12]. The transient behavior starting from  $\bar{\mathbf{m}}_0$  is considered. Let  $M(\tau)$ ,  $X(\tau)$  be the processes describing the evaluation of the discrete and continuous marking, respectively. The following notations are used for  $\mathbf{m}_i \in \mathcal{M}_d$ ,  $\mathbf{x} \in \mathcal{M}_c$  and time  $\tau \geq 0$ :

- $\pi_i(\tau) = \text{Prob}(M(\tau) = \mathbf{m}_i)$  are the discrete state probabilities,
- $H_i(\tau, \mathbf{x})$  is the CDF of the fluid level at fluid places when the discrete state is  $\mathbf{m}_i$ , we have

$$\pi_i(\tau) = \int_0^{B(p_1)} \dots \int_0^{B(p_{|P_c|})} H_i(\tau, (dx_1, \dots, dx_{|P_c|})),$$

- $f(\tau, \mathbf{x}) = \sum_{p_i \in P_d} h_i(\tau, \mathbf{x})$  the fluid density.

For an  $\mathbf{x}$  where  $H_i(\tau, \mathbf{x})$  is continuous  $h_i(\tau, \mathbf{x}) = \frac{\partial}{\partial x_1} \dots \frac{\partial}{\partial x_{|P_c|}} H_i(\tau, \mathbf{x})$  is the probability density of the fluid places. The densities  $h_i(\tau, \mathbf{x})$  are collected in a vector  $\mathbf{h}(\tau, \mathbf{x})$  of length  $|M_d|$ . The dynamic behavior of the system is described by the following set of partial differential equation [12, Theorem 1]

$$\frac{\partial \mathbf{h}(\tau, \mathbf{x})}{\partial \tau} + \sum_{p \in P_c} \frac{\partial (\mathbf{h}(\tau, \mathbf{x}) \mathbf{R}_p(\mathbf{x}))}{\partial x_p} = \mathbf{h}(\tau, \mathbf{x}) \mathbf{Q}(\mathbf{x}). \quad (4)$$

For an  $\mathbf{x}$  where  $H_i(\tau, \mathbf{x})$  is not continuous probability mass develops in various dimensions. These probability masses (e.g., when  $p_1 \in P_c$  is at its lower boundary,  $p_2 \in P_c$  is at its upper boundary and  $p_3 \in P_c$  is between its boundaries), whose number is exponentially increasing with the number of fluid places, are characterized by the boundary equations. Here we avoid the discussion of those boundary equations by referring to [12, Theorem 1], where multi-dimensional masses are considered at the lower boundaries of fluid places.

Results are computed in terms of discrete and continuous markings. We define two functions  $g_d : \mathcal{M}_d \rightarrow \mathbb{R}_{\geq 0}$  and  $g_c : \mathcal{M}_c \rightarrow \mathbb{R}_{\geq 0}$  that indicate the gain or reward with respect to the discrete or continuous state. We assume that the intervals  $\mathcal{M}_c^k$  are defined such that for  $\mathbf{x}, \mathbf{y} \in \mathcal{M}_c^k$   $g_c(\mathbf{x}) = g_c(\mathbf{y})$ . The expectation of the overall gain at time  $\tau$ ,  $G(\tau)$ , is then given by

$$E(G(\tau)) = \sum_{\mathbf{m}_i \in \mathcal{M}_d} \left( \pi_i(\tau) g_d(\mathbf{m}_i) + \int_{x_1} \dots \int_{x_{|P_c|}} g_c(\mathbf{x}) H_i(\tau, (dx_1, \dots, dx_{|P_c|})) \right).$$

To analyze the system numerically, a discretization approach is applied. We introduce a simple first-order scheme following the ideas presented in [3, 8, 12]. Let  $\Delta_p$  the discretization step for place  $p \in P_c$ . We assume that  $B(p)$  is a multiple of  $\Delta_p$  and  $n_p = B(p)/\Delta_p$ . Let  $\Delta = (\Delta_1, \dots, \Delta_{|P_c|})$  be a discretization scheme. Discretization defines a finite state space  $\mathcal{S}_\Delta$  with  $n_\Delta = |M_d| \prod_{p \in P_c} n_p$  states. Each state is defined by a vector  $(u_0, u_1, \dots, u_{|P_c|})$  of length  $1 + |P_c|$  where  $u_0 \in \mathcal{M}_d$  and  $u_p \in \{1, \dots, n_p\}$  for  $p \in P_c$ . States in  $\mathcal{S}_\Delta$  are ordered lexicographically according to their vector representation. Depending on the context, we use the vector representation for states or their number in the state space.

To compute transition rates in the discretized state space different methods exist. We apply a finite volume method and start with the transition rates of the discrete part, as follows

$$q_{ij}^{\mathbf{k}} \triangleq q_{ij}^{(k_1, \dots, k_{|P_c|})} = \frac{1}{\prod_{p \in P_c} \Delta_p} \int_{(k_1-1)\Delta_1}^{k_1\Delta_1} \dots \int_{(k_{|P_c|}-1)\Delta_{|P_c|}}^{k_{|P_c|}\Delta_{|P_c|}} q_{ij}(\mathbf{x}) dx_1, \dots, dx_{|P_c|}, \quad (5)$$

where  $\mathbf{k} = (k_1, \dots, k_{|P_c|})$  is ranging from  $(1, \dots, 1)$  to  $(n_1, \dots, n_{|P_c|})$ . Since the function  $q_{ij}(\mathbf{x})$  is piecewise constant, the integral can be evaluated as a sum. Matrix

$$\bar{\mathbf{Q}} = \begin{pmatrix} \bar{Q}_{1,1} & \dots & \bar{Q}_{1,|M_d|} \\ \vdots & \ddots & \vdots \\ \bar{Q}_{|M_d|,1} & \dots & \bar{Q}_{|M_d|,|M_d|} \end{pmatrix} \quad (6)$$

is a  $n_{\Delta} \times n_{\Delta}$  generator matrix of a Markov chain, where  $\bar{\mathbf{Q}}_{ij} = \text{diag}(q_{ij}^{\mathbf{k}})$  is a diagonal matrix.

For the discretized flow rates define

$$r_{i,\mathbf{k}}^p = \frac{1}{\prod_{p \in P_c} \Delta_p} \int_{(k_1-1)\Delta_1}^{k_1\Delta_1} \cdots \int_{(k_{|P_c|-1}\Delta_{|P_c|})}^{k_{|P_c|}\Delta_{|P_c|}} r_p((\mathbf{m}_i, \mathbf{x})) dx_1, \dots, dx_{|P_c|} \quad (7)$$

for  $\mathbf{m}_i \in \mathcal{M}_d$  as the flow rate of place  $p \in P_c$  when the system is in discrete state  $(\mathbf{m}_i, \mathbf{k}) \in \mathcal{S}_{\Delta}$ . Again the integrals can be evaluated as finite sums since the functions are piecewise constant. For  $\mathbf{k} = (k_1, \dots, k_{|P_c|})$  let  $\mathbf{k} \pm 1_p = (k_1, \dots, k_{p-1}, k_p \pm 1, k_{p+1}, \dots, k_{|P_c|})$ . Observe that  $\mathbf{k} + 1_p$  is not defined if  $k_p = B(p)$  and  $\mathbf{k} - 1_p$  is not defined for  $k_p = 1$ . Now define the flow rates

$$w_i^{\mathbf{k},\mathbf{l}} = \begin{cases} \frac{r_{i,\mathbf{k}}^p}{\Delta_p} & \text{if } \mathbf{l} = \mathbf{k} + 1_p \wedge r_{i,\mathbf{k}}^p > 0, \\ -\frac{r_{i,\mathbf{k}}^p}{\Delta_p} & \text{if } \mathbf{l} = \mathbf{k} - 1_p \wedge r_{i,\mathbf{k}}^p < 0, \\ -\sum_{\ell} \neq \mathbf{k} w_i^{\mathbf{k},\ell} & \text{if } \mathbf{l} = \mathbf{k}, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

and the matrix

$$\bar{\mathbf{W}}_i = \begin{pmatrix} w_i^{(1,\dots,1),(1,\dots,1)} & \cdots & w_i^{(1,\dots,1),(B(1),\dots,B(|P_c|))} \\ \vdots & \ddots & \vdots \\ w_i^{(B(1),\dots,B(|P_c|)),(1,\dots,1)} & \cdots & w_i^{(B(1),\dots,B(|P_c|)),(B(1),\dots,B(|P_c|))} \end{pmatrix}. \quad (9)$$

Then

$$\hat{\mathbf{Q}} = \bar{\mathbf{Q}} + \bar{\mathbf{W}} \text{ where } \bar{\mathbf{W}} = \begin{pmatrix} \bar{\mathbf{W}}_1 & & \\ & \ddots & \\ & & \bar{\mathbf{W}}_{|\mathcal{M}_d|} \end{pmatrix} \quad (10)$$

is the infinitesimal generator matrix of the discretized process such that

$$\frac{d\mathbf{u}(\tau)}{d\tau} = \mathbf{u}(\tau) \hat{\mathbf{Q}} \quad (11)$$

is the system of ordinary differential equations describing the evolution of the discretized process. Let  $\mathbf{u}(\tau)$  be the solution of (11) at time  $\tau$  starting from  $\mathbf{u}(0)$ , which is the discretized version of  $\bar{\mathbf{m}}_0$ .

To approximate  $E(G(\tau))$ , we first define the discretized gain vector for continuous places.

$$g_c^{\mathbf{k}} = \int_{(k_1-1)\Delta_1}^{k_1\Delta_1} \cdots \int_{(k_{|P_c|-1}\Delta_{|P_c|})}^{k_{|P_c|}\Delta_{|P_c|}} g_c(\mathbf{x}) dx_1, \dots, dx_{|P_c|}. \quad (12)$$

Then

$$E(G(\tau)) \approx \sum_{\mathbf{m}_i \in \mathcal{M}_d} \sum_{\mathbf{k}} \mathbf{u}_{(i,\mathbf{k})}(\tau) (g_d(\mathbf{m}_i) + g_c^{\mathbf{k}}), \quad (13)$$

where  $\approx$  indicates the inaccuracy by discretization. For later use we define column vectors  $\mathbf{g}_c, \mathbf{g}_d$  of length  $n_\Delta$  such that  $E(G(\tau)) \approx \hat{G}(\tau) \triangleq \mathbf{u}(\tau) (\mathbf{g}_c + \mathbf{g}_d)$ .

We denote a discretization  $\Delta$  as consistent for  $p \in P_c$ , if  $(b_p^k - b_p^{k-1})/\Delta_p \in \mathbb{N}$  for all intervals. In this case the integrals in (5), (7), (12) can be substituted by sums. A discretization  $\Delta$  is consistent if it is consistent for all  $p \in P_c$ . A refinement of a discretization for place  $p \in P_c$  means to substitute  $\Delta_p$  by  $\Delta_p/i$  for  $i \in \mathbb{N}$ . The number of intervals is increased by a factor  $i$ .  $\Delta/2$  means that every  $\Delta_p$  is substituted by  $\Delta_p/2$ . The number of states is of the discrete process is in this case increased by factor  $2^{|P_c|}$ . If  $\Delta$  is consistent, then every refinement is also consistent.

## 4 Lumping of the Discrete Process

In this section we consider the lumpability of the discretized system developed above. Let  $\sim$  be an equivalence relation on  $\mathcal{S}_\Delta, \hat{\mathcal{S}}_\Delta$  the set of equivalence classes and  $[\mathbf{m}, \mathbf{k}]$  the equivalence class to which  $(\mathbf{m}, \mathbf{k}) \in \hat{\mathcal{S}}_\Delta$  belongs.  $\sim$  is lumpable relation, iff  $\forall \tilde{s}, \tilde{s}' \in \hat{\mathcal{S}}_\Delta, \forall (\mathbf{m}_i, \mathbf{k}), (\mathbf{m}_j, \mathbf{l}) \in [\tilde{s}]$ :

$$\begin{aligned} g_c^k &= g_c^l, \quad g_d(\mathbf{m}_i) = g_d(\mathbf{m}_j), \\ \sum_{(\mathbf{m}_z, \mathbf{y}) \in [\tilde{s}']} \hat{Q}((\mathbf{m}_i, \mathbf{k}), (\mathbf{m}_z, \mathbf{y})) &= \sum_{(\mathbf{m}_z, \mathbf{y}) \in [\tilde{s}']} \hat{Q}((\mathbf{m}_j, \mathbf{l}), (\mathbf{m}_z, \mathbf{y})) \end{aligned} \quad (14)$$

The union of lumpable relations is again a lumpable relation. The lumpable relation with the least number of equivalence classes exists and can be defined as the transitive closure of the union of lumpable partitions. In the sequel we denote this relation by  $\sim$ . It can be computed using partition refinement. Efficient algorithms have been proposed in the past [7, 18] and can also be used in our setting.

Let  $n_\Delta$  be the number of states and  $\tilde{n}_\Delta$  the number of equivalence classes of  $\sim$ . The equivalence relation can be represented by a so-called collector matrix [1] which is a  $n_\Delta \times \tilde{n}_\Delta$  matrix  $\mathbf{V}$  with  $\mathbf{V}(j, i) = 1$  if  $j \in [i]$  and 0 if  $j \notin [i]$ . Matrix  $\mathbf{V}$  contains one element equal to 1 in each row and at least one element equal to 1 in each column. A distributor matrix is defined as an  $\tilde{n}_\Delta \times n_\Delta$  matrix  $\mathbf{W} = \overline{(\mathbf{V})^T}$  where the overline means that the rows of the transposed matrix  $\mathbf{V}$  are normalized to 1. Then  $\tilde{\mathbf{Q}} = \mathbf{W}\hat{\mathbf{Q}}\mathbf{V}$  is the  $\tilde{n}_\Delta \times \tilde{n}_\Delta$  matrix of the lumped system. It is easy to show [1, 15] that the relation  $\hat{\mathbf{Q}}\mathbf{V} = \mathbf{V}\tilde{\mathbf{Q}}$  holds in this case. Furthermore, define  $\tilde{\mathbf{g}}_c^T = \mathbf{g}_c^T \mathbf{V}$  and  $\tilde{\mathbf{g}}_d^T = \mathbf{g}_d^T \mathbf{V}$  which implies, due to the lumpability conditions,  $\mathbf{g}_c = \mathbf{V}\tilde{\mathbf{g}}_c$  and  $\mathbf{g}_d = \mathbf{V}\tilde{\mathbf{g}}_d$ . Then

$$\frac{\partial \tilde{\mathbf{u}}(\tau)}{\partial \tau} = \tilde{\mathbf{u}}(\tau) \tilde{\mathbf{Q}} \quad (15)$$

are the ordinary differential equations for the lumped system. The initial condition is  $\tilde{\mathbf{u}}(0) = \mathbf{u}(0)\mathbf{V}$ .  $E(G(\tau)) \approx \tilde{G}(\tau) = \tilde{\mathbf{u}}(\tau) (\tilde{\mathbf{g}}_c + \tilde{\mathbf{g}}_d)$  is the result of the lumped system.

**Theorem 1.** *If the lumped system has been generated according to some lumpable equivalence relation  $\sim$ , then  $\tilde{G}(\tau) = \hat{G}(\tau)$ .*

*Proof.* We show here that the forward Euler method with time step  $\delta$  applied to (11) and (15) yields the same results. Since the Euler method converges for  $\delta \rightarrow 0$  towards the exact solution, both sets of ordinary differential equations converge towards the same solution. Let  $\mathbf{u}^k = \mathbf{u}(k \cdot \delta)$  and  $\tilde{\mathbf{u}}^k = \tilde{\mathbf{u}}(k \cdot \delta)$ , then  $\tilde{\mathbf{u}}^0 = \mathbf{u}^0 \mathbf{V}$ . We show by induction that the relation holds for all  $k = 0, 1, \dots$

Assume that  $\tilde{\mathbf{u}}^k = \mathbf{u}^k \mathbf{V}$ , then the  $(k+1)$ th vector is computed by the forward Euler scheme as

$$\tilde{\mathbf{u}}^{k+1} = \tilde{\mathbf{u}}^k + \delta \tilde{\mathbf{u}}^k \tilde{\mathbf{Q}} = \mathbf{u}^k \mathbf{V} + \delta \mathbf{u}^k \mathbf{V} \tilde{\mathbf{Q}} = \left( \mathbf{u}^k + \delta \mathbf{u}^k \tilde{\mathbf{Q}} \right) \mathbf{V} = \mathbf{u}^{k+1} \mathbf{V}$$

and

$$\tilde{G}(k \cdot \delta) = \tilde{\mathbf{u}}^k (\tilde{g}_c + \tilde{g}_d) = \mathbf{u}^k \mathbf{V} (\tilde{g}_c + \tilde{g}_d) = \mathbf{u}^k (g_c + g_d) = \hat{G}(k \cdot \delta),$$

which completes the proof.  $\square$

We now consider refinements of consistent partitions. Let  $\Delta$  be a partition which is consistent for  $p \in P_c$  and let  $\Delta'$  be a partition that results from  $\Delta$  by substituting  $\Delta_p$  by  $\Delta_p/2$ , then each state  $(\mathbf{m}, \mathbf{k}) \in \mathcal{S}_\Delta$  is represented by two states  $(\mathbf{m}, \mathbf{k}^-), (\mathbf{m}, \mathbf{k}^+) \in \mathcal{S}_{\Delta'}$  where  $\mathbf{k}^- = (k_1, \dots, k_{p-1}, 2k_p-1, k_{p+1}, \dots, k_{|P_c|})$  and  $\mathbf{k}^+ = (k_1, \dots, k_{p-1}, 2k_p, k_{p+1}, \dots, k_{|P_c|})$ . The non-diagonal elements of matrix  $\hat{\mathbf{Q}}_{\Delta'}$  can be derived from the elements of  $\hat{\mathbf{Q}}_\Delta$  as follows:

$$\begin{aligned} \hat{\mathbf{Q}}_{\Delta'}((\mathbf{m}, \mathbf{k}^+), (\mathbf{m}', \mathbf{k}^+)) &= \hat{\mathbf{Q}}_{\Delta'}((\mathbf{m}, \mathbf{k}^-), (\mathbf{m}', \mathbf{k}^-)) = \hat{\mathbf{Q}}_\Delta((\mathbf{m}, \mathbf{k}), (\mathbf{m}', \mathbf{k})) \\ \hat{\mathbf{Q}}_{\Delta'}((\mathbf{m}, \mathbf{k}^+), (\mathbf{m}, \mathbf{l}^+)) &= \hat{\mathbf{Q}}_{\Delta'}((\mathbf{m}, \mathbf{k}^-), (\mathbf{m}, \mathbf{l}^-)) = \hat{\mathbf{Q}}_\Delta((\mathbf{m}, \mathbf{k}), (\mathbf{m}, \mathbf{l})) \\ \hat{\mathbf{Q}}_{\Delta'}((\mathbf{m}, \mathbf{k}^+), (\mathbf{m}, \mathbf{k}^-)) &= -\frac{\min(r_{\mathbf{m}, \mathbf{k}^+}^p, 0)}{2\Delta_p} \\ \hat{\mathbf{Q}}_{\Delta'}((\mathbf{m}, \mathbf{k}^-), (\mathbf{m}, \mathbf{k}^+)) &= \frac{\max(r_{\mathbf{m}, \mathbf{k}^-}^p, 0)}{2\Delta_p} \\ \hat{\mathbf{Q}}_{\Delta'}((\mathbf{m}, \mathbf{k}^+), (\mathbf{m}, (\mathbf{k}^+ + 1_p))) &= \frac{\max(r_{\mathbf{m}, \mathbf{k}^+}^p, 0)}{2\Delta_p} \\ \hat{\mathbf{Q}}_{\Delta'}((\mathbf{m}, \mathbf{k}^-), (\mathbf{m}, (\mathbf{k}^- - 1_p))) &= -\frac{\min(r_{\mathbf{m}, \mathbf{k}^-}^p, 0)}{2\Delta_p} \end{aligned} \tag{16}$$

for  $\mathbf{m} \neq \mathbf{m}'$  and  $\mathbf{l} \neq \mathbf{k}$  and  $\mathbf{l} \neq \mathbf{k} \pm 1_p$ . All remaining non-diagonal matrix entries are 0. The diagonal elements  $\hat{\mathbf{Q}}_{\Delta'}((\mathbf{m}, \mathbf{k}^+), (\mathbf{m}, \mathbf{k}^+)), \hat{\mathbf{Q}}_{\Delta'}((\mathbf{m}, \mathbf{k}^-), (\mathbf{m}, \mathbf{k}^-))$  are chosen to yield row sum 0 in each row. For the lumped system let  $[\mathbf{m}, \mathbf{l}]$  be the equivalence class to which state  $(\mathbf{m}, \mathbf{l})$  belongs. Each equivalence class  $[\mathbf{m}, \mathbf{k}]$  is split into two classes  $[\mathbf{m}, \mathbf{k}^-], [\mathbf{m}, \mathbf{k}^+]$ . Matrix  $\tilde{\mathbf{Q}}_{\Delta'}$  is defined by (16) where  $\hat{\mathbf{Q}}_{\Delta'}((\mathbf{m}, \mathbf{k}), (\mathbf{m}', \mathbf{k}'))$  is substituted by  $\tilde{\mathbf{Q}}_{\Delta'}((\mathbf{m}, \mathbf{k}), (\mathbf{m}', \mathbf{k}'))$ .

**Theorem 2.** *Let  $\Delta'$  be a refinement of  $\Delta$ . If  $\tilde{\mathbf{Q}}_\Delta$  results from  $\hat{\mathbf{Q}}_\Delta$  by a lumpable equivalence relation  $\sim$ , then  $\tilde{\mathbf{Q}}_{\Delta'}$  results from  $\hat{\mathbf{Q}}_{\Delta'}$  by a lumpable equivalence relation  $\sim'$  where  $(\mathbf{m}, \mathbf{k}) \sim (\mathbf{m}', \mathbf{l}) \Rightarrow ((\mathbf{m}, \mathbf{k}^-) \sim' (\mathbf{m}', \mathbf{l}^-)) \wedge ((\mathbf{m}, \mathbf{k}^+) \sim' (\mathbf{m}', \mathbf{l}^+))$ .*



*Proof.* Consider a pair of states  $(\mathbf{m}, \mathbf{k}^+) \sim' (\mathbf{m}', \mathbf{l}^+)$ . Then it holds that  $(\mathbf{m}, \mathbf{k}) \sim (\mathbf{m}', \mathbf{l})$ . We have to show that the sum of rates into the state from each equivalence class  $[\mathbf{m}'', \mathbf{y}^\pm]$  is identical for both states. All rates out of  $(\mathbf{m}, \mathbf{k}^+)$  ( $(\mathbf{m}', \mathbf{l}^+)$ , resp.) that do not result from a flow into or out of place  $p$  are identical to the corresponding rates out of state  $(\mathbf{m}, \mathbf{k})$  ( $(\mathbf{m}', \mathbf{l})$ , resp.). Thus, the rates in the first two cases of (16) are identical, which means that we only have to consider the remaining cases.

Since the discretization is consistent,  $r_{\mathbf{m}, \mathbf{k}}^p = r_{\mathbf{m}', \mathbf{l}}^p$  has to hold which implies that  $\frac{r_{\mathbf{m}, \mathbf{k}}^p}{2\Delta'} = \frac{r_{\mathbf{m}', \mathbf{l}}^p}{2\Delta'}$  and lumpability is transferred from  $\sim$  to  $\sim'$ .

The proof for the other case,  $(\mathbf{m}, \mathbf{k}^-) \sim' (\mathbf{m}', \mathbf{l}^-)$  uses similar arguments.  $\square$

If  $r_{\mathbf{m}, \mathbf{k}}^p = 0$  for all  $p \in P_c$ , then  $\sim'$  can be extended by joining the equivalence classes  $[\mathbf{m}, \mathbf{k}^-]$  and  $[\mathbf{m}', \mathbf{k}^+]$ . This can be seen by noticing that the flow between  $(\mathbf{m}, \mathbf{k}^+)$  and  $(\mathbf{m}, \mathbf{k}^-)$  is 0 in this case.

The results presented in this section suggest the following lumping approach:

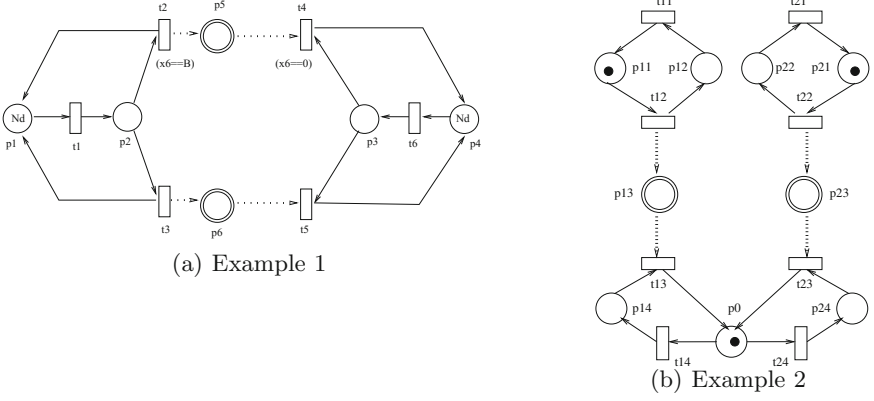
1. Find the coarsest consistent discretization  $\Delta$ .
2. Build matrix  $\hat{Q}_\Delta$ .
3. Compute the largest lumpable equivalence relation using partition refinement.
4. Refine the discretization and generate the lumped matrices using (16) such that the discretization error remains small enough. The discretization error for the finite volume method, that is applied, can be estimated using standard methods for the numerical solution of partial differential equations [14].
5. Compute the results by solving the set of ordinary differential equations (11) resulting from the lumped system.

The refinement described here refines  $\Delta_p$  for all fluid places in the same way. It is, of course, possible to restrict the refinement to some fluid places only.

The whole approach works well, if the transition and flow rates are defined in such a way that a coarse consistent discretization exists. Otherwise the lumpable partition has to be computed for the fine discretization that is used for analysis. Such an approach is only useful for large time horizons  $\tau$ , for which the solution should be computed.

*Example 1.* We consider the simple net shown in Fig. 1(a). It describes a source and a sink which have  $N_d$  operational modes. In mode  $i \in \{0, \dots, N_d\}$ , indicated by  $i$  tokens on place  $p_2$  or  $p_3$ , fluid is produced respectively consumed. Production of fluid is described by the transitions  $t_2, t_3$ , consumption by the transitions  $t_4, t_5$ . In mode 0 the consumer or producer are switched off, which means that the transitions are not enabled. The net contains two fluid buffers modeled by the fluid places  $p_5$  and  $p_6$ . Both buffers have the same capacity  $B$  but buffer  $p_6$  is always filled and emptied first. This implies that the behavior is non-symmetric, the fluid densities differ for both fluid places.

The system becomes lumpable if the input transition  $t_2$  and  $t_3$  as well as the output transitions  $t_3$  and  $t_4$  of the buffers have identical parameters which depend only on the sum of fluid in fluid places and not on the individual fluid

**Fig. 1.** Example FSPNs.

levels  $x_5$  and  $x_6$ . For the firing rates this means that  $t_2$  is enabled if  $m_2 > 0$ ,  $x_5 < B$  and  $x_6 = B$ . The firing rate is then given by  $F(t_2, (\mathbf{m}, x_5, x_6)) = \lambda(\mathbf{m}, x_5 + x_6)$ . Transition  $t_3$  is enabled if  $m_2 > 0$  and  $x_6 < B$ . The firing rate equals  $F(t_3, (\mathbf{m}, x_5, x_6)) = \lambda(\mathbf{m}, x_5 + x_6)$ . Similarly,  $t_4$  is enabled with firing rate  $F(t_4, (\mathbf{m}, x_5, x_6)) = \mu(\mathbf{m}, x_5 + x_6)$  if  $m_3 > 0$ ,  $x_5 > 0$  and  $x_6 = 0$  and  $t_5$  is enabled with rate  $F(t_5, (\mathbf{m}, x_5 + x_6)) = \mu(\mathbf{m}, x_5 + x_6)$  if  $m_3 > 0$  and  $x_6 > 0$ .  $\lambda(\cdot)$  and  $\mu(\cdot)$  are positive functions. The flow rates underlie similar restrictions. We have  $R((t_2, p_5), (\mathbf{m}, x_5, x_6)) = \nu(\mathbf{m}, x_5 + x_6)$ ,  $R((t_3, p_6), (\mathbf{m}, x_5, x_6)) = \nu(\mathbf{m}, x_5 + x_6)$ ,  $R((t_4, p_5), (\mathbf{m}, x_5, x_6)) = \xi(\mathbf{m}, x_5 + x_6)$  and  $R((t_5, p_6), (\mathbf{m}, x_5, x_6)) = \xi(\mathbf{m}, x_5 + x_6)$ .  $\nu(\cdot)$  and  $\xi(\cdot)$  are in our setting piecewise constant positive functions. Lumping additionally requires that all results are only based on the discrete marking and the sum of the content of the fluid places,  $x_5 + x_6$ .

If the system is lumpable, in some situations only the sum of fluid of the fluid places and not the individual fluid levels have to be considered in the state space. This results in a significant state space reduction. Table 1 shows some state space sizes for the detailed and lumped system of equations. Parameter  $N_d$  describes the number of tokens in the places  $p_1$  and  $p_4$  in the initial marking and  $N_c$  is the number of discretization intervals for the fluid places.

**Table 1.** State space sizes for the first example net.

$N_d$	$N_c$	Original	Lumped	$N_d$	$N_c$	Original	Lumped
1	6	140	40	2	6	318	93
1	11	480	80	2	11	1083	183
1	21	1760	160	2	21	3963	363
1	51	10400	400	2	51	23403	903
1	101	40800	800	2	101	91803	1803

## 5 Lumping of the FSPN Matrices

Lumping, as presented in the previous section, is done at the state space of the discrete process. Alternatively several approaches have been proposed to perform lumping compositionally [2, 11] or based on symmetries at the net level [4]. In principle similar approaches can be developed for FSPNs after defining compositional or colored nets. However, here we consider an intermediate step by defining and computing lumpable partitions at the levels of the matrices  $Q(\mathbf{x})$  and  $R_p(\mathbf{x})$  ( $p \in P_c$ ), defined at the beginning of Sect. 3.

In Theorem 3, we define an equivalence relation  $\sim$  that relates states  $\bar{m} = (\mathbf{m}, \mathbf{x})$  and  $\bar{m}' = (\mathbf{m}', \mathbf{x}')$ . We can restrict the equivalence relation to the discrete and continuous parts of the state description. Thus, if  $(\mathbf{m}, \mathbf{x}) \sim (\mathbf{m}', \mathbf{x}')$ , then  $\mathbf{m} \sim_d \mathbf{m}'$  and  $\mathbf{x} \sim_c \mathbf{x}'$ . However, in general, the discrete and continuous parts of the relation are not independent. I.e.,  $(\mathbf{m}, \mathbf{x}) \sim (\mathbf{m}', \mathbf{x}') \Rightarrow \mathbf{m} \sim_d \mathbf{m}' \wedge \mathbf{x} \sim_c \mathbf{x}'$  but the other direction  $\Leftarrow$  usually will not hold. We furthermore assume that  $\sim_c$  defines equivalence classes on the set of continuous places. This is a restriction because we consider only continuous states as equivalent which are identical up to the ordering of components in the state vector. Therefore we define permutations  $perm$  that permute the indices of continuous places. With a slight misuse of notation we may use  $perm$  on the state vector of the continuous part, e.g.  $perm(\mathbf{x})$  describes the renumbering of the positions in the vector according to  $perm$ . If  $\mathbf{x} \sim_c \mathbf{x}'$ , then there exists some permutation  $perm$  of continuous places, such that  $\mathbf{x}' = perm(\mathbf{x})$ . Thus,  $\sim_c$  induces an equivalence relation on  $P_c$  and  $\mathbf{x} \sim_c \mathbf{x}'$  implies that  $\mathbf{x}'$  results from  $\mathbf{x}$  by reordering equivalent places. We denote by  $\mathcal{P}_c$  the set of permutations that permute equivalent continuous places. Since  $\sim_p$  is an equivalence relation if  $perm \in \mathcal{P}_c$ , then  $perm^{-1} \in \mathcal{P}_c$  and if  $perm, perm' \in \mathcal{P}_c$ , then  $perm \circ perm' \in \mathcal{P}_c$  where  $perm \circ perm'$  is the concatenation of permutations  $perm$  and  $perm'$ .

**Theorem 3.** *An equivalence relation  $\sim$  defines a lumpable partition on the state space of a FSPN, if for all equivalence classes  $[\mathbf{m}, \mathbf{x}], [\mathbf{m}', \mathbf{x}']$  and for all  $(\mathbf{m}_i, \mathbf{x}_i), (\mathbf{m}_j, \mathbf{x}_j) \in [\mathbf{m}, \mathbf{x}]$  the following relations hold:*

$$\begin{aligned} &\exists \text{ a set of permutations } \mathcal{P}_c \text{ on } P_c \text{ such that } \mathbf{x}_i = perm(\mathbf{x}_j) \text{ for some } perm \in \mathcal{P}_c, \\ &\sum_{(\mathbf{m}_k, \mathbf{x}_k) \in [\mathbf{m}', \mathbf{x}']} q_{ik}(\mathbf{x}_i) = \sum_{(\mathbf{m}_k, \mathbf{x}_k) \in [\mathbf{m}', \mathbf{x}']} q_{jk}(\mathbf{x}_j), \quad g_d(\mathbf{m}_i) = g_d(\mathbf{m}_j), \quad g_c(\mathbf{x}_i) = g_c(\mathbf{x}_j), \\ &\forall p \in P_c : r_p(\mathbf{m}_i, \mathbf{x}_i) = r_{perm(p)}(\mathbf{m}_j, \mathbf{x}_j), \quad B(p) = B(perm(p)) \text{ for } \mathbf{x}_j = perm(\mathbf{x}_i). \end{aligned} \quad (17)$$

*Proof.* We have to prove that (14) holds for arbitrary discretizations  $\Delta$  where  $\Delta_p = \Delta_{p'}$  if  $p = perm(p')$  for some  $perm \in \mathcal{P}_\sim$ .

Now let  $\Delta$  be some discretization of the above form and  $\hat{Q}$  the corresponding rate matrix.  $\sim$  induces an equivalence relation  $\sim$  on the discrete state space such that  $(\mathbf{m}, \mathbf{k}) \sim (\mathbf{m}', \mathbf{k}')$  iff  $\mathbf{m} \sim \mathbf{m}'$  and  $\mathbf{k} = perm(\mathbf{k}')$  for some  $perm \in \mathcal{P}_\sim$ .

We first consider equivalence of the rewards. For the continuous and discrete reward we have by definition of  $\sim$  that  $g_d(\mathbf{m}_i) = g_d(\mathbf{m}_j)$  and  $g_c(\mathbf{x}_i) = g_c(\mathbf{x}_j)$

holds for  $\mathbf{m}_i \sim \mathbf{m}_j$  and  $\mathbf{x}_i \sim \mathbf{x}_j$ . For continuous places this implies that also the integrals in (12) are identical for  $\mathbf{k}$  and  $\mathbf{k}'$  which implies  $g_c^{\mathbf{k}} = g_c^{\mathbf{k}'}$ .

Now consider the lumpability condition on the sums of rates, namely

$$\sum_{(\mathbf{n}', \mathbf{l}') \in [(\mathbf{n}, \mathbf{l})]} \widehat{Q}((\mathbf{m}, \mathbf{k}), (\mathbf{n}', \mathbf{l}')) = \sum_{(\mathbf{n}', \mathbf{l}') \in [(\mathbf{n}, \mathbf{l})]} \widehat{Q}((\mathbf{m}', \mathbf{k}'), (\mathbf{n}', \mathbf{l}'))$$

for some equivalence class  $[(\mathbf{n}, \mathbf{l})]$  of  $\sim$  and  $(\mathbf{m}, \mathbf{k}) \sim (\mathbf{m}', \mathbf{k}')$ . The rates can result from a change of the discrete marking by firing a transition (collected in  $\overline{Q}$ ) or from the discretized continuous flow (collected in  $\overline{W}$ ). In the former case the rate for  $(\mathbf{m}, \mathbf{k}) \notin [(\mathbf{n}, \mathbf{l})]$  is given by

$$\begin{aligned} \sum_{(\mathbf{n}', \mathbf{l}') \in [(\mathbf{n}, \mathbf{k})]} \widehat{Q}((\mathbf{m}, \mathbf{k}), (\mathbf{n}', \mathbf{l}')) &= \\ \sum_{(\mathbf{n}', \mathbf{l}') \in [(\mathbf{n}, \mathbf{l})]} \int_{(k_1-1)\Delta_1}^{k_1\Delta_1} \cdots \int_{(k_{|P_c|-1}\Delta_{|P_c|})}^{k_{|P_c|}\Delta_{|P_c|}} q_{\mathbf{m}, \mathbf{n}'}(\mathbf{x}) dx_1, \dots, dx_{|P_c|} &= \\ \int_{(k_1-1)\Delta_1}^{k_1\Delta_1} \cdots \int_{(k_{|P_c|-1}\Delta_{|P_c|})}^{k_{|P_c|}\Delta_{|P_c|}} \sum_{(\mathbf{n}', \mathbf{l}') \in [(\mathbf{n}, \mathbf{l})]} q_{\mathbf{m}, \mathbf{n}'}(\mathbf{x}) dx_1, \dots, dx_{|P_c|} &= \\ \sum_{(\mathbf{n}', \mathbf{l}') \in [(\mathbf{n}, \mathbf{l})]} \int_{(k_1-1)\Delta_1}^{k_1\Delta_1} \cdots \int_{(k_{|P_c|-1}\Delta_{|P_c|})}^{k_{|P_c|}\Delta_{|P_c|}} q_{\mathbf{m}', \mathbf{n}'}(\text{perm}(\mathbf{x})) dx_1, \dots, dx_{|P_c|} &= \\ \sum_{(\mathbf{n}', \mathbf{l}') \in [(\mathbf{n}, \mathbf{l})]} \widehat{Q}((\mathbf{m}', \mathbf{k}'), (\mathbf{n}', \mathbf{l}')) \end{aligned}$$

Observe that the continuous part is not modified by firing the discrete transition but discretized vectors may differ due to a permutation of equivalent continuous places. Sum and integrals can be interchanged due to Fubini's theorem. If the identity holds for all equivalence classes  $[(\mathbf{n}, \mathbf{l})] \neq [(\mathbf{m}, \mathbf{k})]$  then it also holds for  $[(\mathbf{m}, \mathbf{k})]$  because  $\overline{Q}$  has zero row sums.

If the state changes due to the discretized continuous flow, the discrete remains and we have for the flow according to place  $p \in P_c$ .

$$\begin{aligned} \sum_{(\mathbf{n}', \mathbf{k}) \in [(\mathbf{n}, \mathbf{k})]} \widehat{Q}((\mathbf{m}, \mathbf{k}), (\mathbf{n}', \mathbf{k} \pm 1_p)) &= \\ \left| \frac{1}{\Delta_p \prod_{p' \in P_c} \Delta_{p'}} \sum_{(\mathbf{n}', \mathbf{l}') \in [(\mathbf{n}, \mathbf{k} \pm 1_p)]} \int_{(k_1-1)\Delta_1}^{k_1\Delta_1} \cdots \int_{(k_{|P_c|-1}\Delta_{|P_c|})}^{k_{|P_c|}\Delta_{|P_c|}} r_p((\mathbf{m}, \mathbf{x})) dx_1, \dots, dx_{|P_c|} \right| &= \\ \left| \frac{1}{\Delta_p \prod_{p' \in P_c} \Delta_{p'}} \int_{(k_1-1)\Delta_1}^{k_1\Delta_1} \cdots \int_{(k_{|P_c|-1}\Delta_{|P_c|})}^{k_{|P_c|}\Delta_{|P_c|}} \sum_{(\mathbf{n}', \mathbf{l}') \in [(\mathbf{n}, \mathbf{k} \pm 1_p)]} r_p((\mathbf{m}, \mathbf{x})) dx_1, \dots, dx_{|P_c|} \right| &= \\ \left| \frac{1}{\Delta_p \prod_{p' \in P_c} \Delta_{p'}} \sum_{(\mathbf{n}', \mathbf{l}') \in [(\mathbf{n}, \mathbf{k} \pm 1_p)]} \int_{(k_1-1)\Delta_1}^{k_1\Delta_1} \cdots \int_{(k_{|P_c|-1}\Delta_{|P_c|})}^{k_{|P_c|}\Delta_{|P_c|}} r_{\text{perm}(p)}((\mathbf{m}', \text{perm}(\mathbf{x}))) dx_1, \dots, dx_{|P_c|} \right| &= \\ \sum_{(\mathbf{n}', \mathbf{k}) \in [(\mathbf{n}, \mathbf{k})]} \widehat{Q}((\mathbf{m}', \mathbf{k}'), (\mathbf{n}', \mathbf{k}' \pm 1_{\text{perm}(p)})) \end{aligned}$$

$\pm$  equals  $+$  if the resulting value is positive, otherwise it equals  $-$ .

Thus, the lumpability conditions hold for any discretization  $\Delta$  which completes the proof.  $\square$

Partition  $\sim$  can be computed by partition refinement. The corresponding algorithm will be briefly outlined in the following steps.

1. Generate the discrete state space  $\mathcal{M}_d$  (by assumption this can be done by neglecting continuous places) and set  $k = 0$ .
2. Define an initial equivalence relation  $\sim_d^0$  on  $\mathcal{M}_d$  by  $\mathbf{m} \sim_d^0 \mathbf{m}'$  for  $\mathbf{m}, \mathbf{m}' \in \mathcal{M}_d$ , iff  $g_d(\mathbf{m}) = g_d(\mathbf{m}')$ .
3. Define an initial equivalence relation  $\sim_c^0$  by  $p \sim_c^0 p'$  for  $p, p' \in P_c$  and for all  $\mathbf{x} \in \mathcal{M}_c$ , iff  $B(p) = B(p')$  and  $g_c(\mathbf{x}) = g_c(\mathbf{x}_{p \leftrightarrow p'})$  where  $\mathbf{x}_{p \leftrightarrow p'}$  results from  $\mathbf{x}$  by exchanging the positions for  $p$  and  $p'$ . Let  $Perm^0$  be the set of all permutations that permute the indices of equivalent places from  $P_c$ .
4. Partition refinement of  $\sim_d^k$ : for all equivalence classes  $[\mathbf{m}]$  split  $[\mathbf{m}]$  into new equivalence classes  $[\mathbf{m}^1], \dots, [\mathbf{m}^L]$  until  $\sum_{\mathbf{m}_k \in [\mathbf{m}^l]} q_{i,k}(\mathbf{x}) = \sum_{\mathbf{m}_k \in [\mathbf{m}^l]} q_{j,k}(\text{perm}(\mathbf{x}))$  holds for all  $\mathbf{m}_i, \mathbf{m}_j \in [\mathbf{m}^l]$  ( $l = 1, \dots, L$ ), for all  $[\mathbf{m}'] \in \sim_d^k$ , for all  $\mathbf{x} \in \mathcal{M}_c$  and some  $\text{perm} \in Perm^k$ , add equivalence classes  $[\mathbf{m}^1], \dots, [\mathbf{m}^L]$  to  $\sim_d^{k+1}$ .
5. Partition refinement of  $\sim_c^k$ : for all equivalence classes  $[q]$  of  $\sim_c^k$  split  $[q]$  into equivalence classes  $[q^1], \dots, [q^K]$  until for all  $p, p' \in [q^k]$  ( $k = 1, \dots, K$ ), all equivalence classes  $[\mathbf{m}]$  of  $\sim_d^{k+1}$ , exist  $\mathbf{m}_i, \mathbf{m}_j \in [\mathbf{m}]$   $r_p(\mathbf{m}_i, \mathbf{x}) = r_{p'}(\mathbf{m}_j, \text{perm}(\mathbf{x}))$  for all  $\mathbf{x}$  where  $\text{perm}$  is the permutation that assures in step 4 that  $\mathbf{m}_i$  and  $\mathbf{m}_j$  are equivalent add equivalence classes  $[q^1], \dots, [q^K]$  to  $\sim_c^{k+1}$ .
6. If  $\sim_d^k \equiv \sim_d^{k-1}$  and  $\sim_c^k \equiv \sim_c^{k-1}$ , then stop (a lumpable partition has been found), else set  $k = k + 1$  and continue with step 4.

Some remarks should be given for the outlined algorithm. The algorithm eventually terminates because in each iteration at least one new equivalence class is generated for the states in  $\mathcal{M}_d$  or the places in  $P_c$  and the finest equivalence relation is the identity relation where each equivalence class contains a single discrete state or continuous place, respectively. Since the number of places and the number of discrete markings are finite by assumption, the algorithm will stop. Knowledge of  $\sim_d$  and  $\sim_c$  is not sufficient to define the lumpable relation, additionally we need the relation between equivalent discrete states and the corresponding permutation of equivalent continuous places (see step 4 of the algorithm). The partition refinement in step 4 requires that the rates are identical for all  $\mathbf{x} = (x_1, \dots, x_{|P_c|})$  where  $x_p \in [0, B(p)]$ . To check this algorithmically an appropriate specification of the rates  $q_{ik}(\mathbf{x})$  is necessary which is also required for the specification of rates depending on the filling of fluid places.

It should be noted that the lumpability conditions for the discrete and continuous part are not symmetric. For the discrete part we define lumping at the state level of the stochastic process, whereas for the continuous part lumping is defined for symmetric places which is more restrictive. Consequently, a coarser lumpable partition may exist which cannot be found by the outlined algorithm above but can be computed by partition refinement of an adequately discretized process (see e.g. Example 1).

*Example 2.* The second example is a symmetric FSPN which is shown in Fig. 1(b) in a version with two components. Places with names  $pk_i$  belong to component  $k$ . Each component models a switched source that produces fluid for

a continuous place  $pk3$ . A single consumer exists which is idle if a token resides on place  $p0$ . If transition  $tk4$  fires, the consumer changes its state to a state where fluid from place  $pk3$  is consumed. If transition  $tk3$  fires, the consumer goes back to the idle state. The model can be defined for  $>2$  components in exactly the same way.

To allow state space reduction by lumpability, the components and the consumer have to show a symmetric behavior. We define the corresponding conditions for the case of  $K$  components. Let  $\mathcal{P}_K$  be the set of all permutations of the numbers 1 through  $K$ . We use the notation  $perm(\mathbf{m})$  and  $perm(\mathbf{x})$  to indicate the application of permutation  $perm$  on the vector which means that the vector components belonging to the corresponding places are exchanged. I.e., if  $perm(k) = l$ , then  $pki$  becomes  $pli$  and  $tki$  becomes  $tli$ . Observe that discrete places are mapped on discrete places and continuous places on continuous places. The following equalities have to hold for all  $k, l \in \{1, \dots, K\}$ , all  $perm \in \mathcal{P}_K$  where  $perm(k) = l$ , all transitions  $tki$  and all places  $pki$  to assure lumpability.

$$F(tki, (\mathbf{m}, \mathbf{x})) = F(tli, (perm(\mathbf{m}, \mathbf{x}))), \quad r_{pki}(\mathbf{m}, \mathbf{x}) = r_{pli}(perm(\mathbf{m}, \mathbf{x})), \\ g_c^x = g_c^{perm(\mathbf{x})} \quad \text{and} \quad g_d(\mathbf{m}) = g_d(perm(\mathbf{m})).$$

Additionally, the bounds for the continuous places have to be identical and the same discretization has to be applied for all continuous places. Observe that due to an appropriate definition of marking dependent transition rates, the firing of transitions may depend on the filling of continuous places. For example  $tk4$  may only have a non-zero rate if  $pk3$  includes enough fluid and the rate of  $tk3$  may grow if  $pk3$  becomes empty. Let  $n_c$  be the number of discrete intervals resulting from the discretization of the continuous state of each continuous place. Then  $\mathcal{S}_\Delta$  contains  $(2K + 1)(2n_c)^K$  states.

A lumpable equivalence relation  $\sim$  can be generated by defining two states as equivalent if one can be transformed into the other by a permutation  $perm \in \mathcal{P}_K$ . Thus,  $(\mathbf{m}, \mathbf{x}) \sim (\mathbf{m}', \mathbf{x}')$  if  $perm(\mathbf{m}, \mathbf{x}) = (\mathbf{m}', \mathbf{x}')$  for some  $perm \in \mathcal{P}_K$ . For the reduced state space, we do not have to distinguish the identity of a component, we only have to consider the number of equivalent components which are in a specific state. This is well known from symmetry exploitation in SPNs [4]. For our example this means that if a token resides at place  $p0$ , then all components are identical and the number of states is reduced from  $(2n_c)^K$  to  $\binom{2n_c + K - 1}{K}$ . If a token resides at some place  $pk4$ , then the identity of the place is not relevant and the remaining components are identical such that the number of states is reduced from  $K(2n_c)^K$  to  $2n_c * \binom{2n_c + K - 2}{K - 1}$ . Overall the lumped state space  $\tilde{\mathcal{S}}_\Delta$  includes  $\binom{2n_c + K - 1}{K} + 2n_c * \binom{2n_c + K - 2}{K - 1}$  states. For larger values of  $K$  the reduction becomes significant but the state space still remains large if  $n_c$  is large.

## 6 Conclusions

Lumping proved to be an efficient tool in the analysis of discrete state systems such as CTMCs and SPNs. In this work we extended the concept of lumping

to FSPNs which are hybrid (discrete and continuous state) systems. The fundamental approach behind this extension is to map the hybrid system to a discrete state one and apply the available lumping relations for the discrete system.

To pursue this approach we presented a discretization of FSPNs and elaborated on the refinement of the discretization step. We showed that the refinement maintains the lumping relation, which is important for utilizing the fact that asymptotic behavior of the discrete system tends to the hybrid one as the discretization step tends to zero. Additionally we presented an approach where the lumping of the continuous part is based on the symmetry among continuous places which is less general but may be proved by generating only the discrete state space without building the discretized continuous part.

## References

1. Buchholz, P.: Exact and ordinary lumpability in finite Markov chains. *J. Appl. Probab.* **31**, 59–75 (1994)
2. Buchholz, P.: A notion of equivalence for stochastic Petri nets. In: Michelis, G., Diaz, M. (eds.) ICATPN 1995. LNCS, vol. 935, pp. 161–180. Springer, Heidelberg (1995). doi:[10.1007/3-540-60029-9\\_39](https://doi.org/10.1007/3-540-60029-9_39)
3. Chen, D., Hong, Y., Trivedi, K.S.: Second-order stochastic fluid models with fluid-dependent flow rates. *Perform. Eval.* **49**(1/4), 341–358 (2002)
4. Chiola, G., Dutheillet, C., Franceschinis, G., Haddad, S.: Stochastic well-formed colored nets and symmetric modeling applications. *IEEE Trans. Comput.* **42**(11), 1343–1360 (1993)
5. Ciardo, G., Nicol, D.M., Trivedi, K.S.: Discrete-event simulation of fluid stochastic Petri nets. *IEEE Trans. Softw. Eng.* **25**(2), 207–217 (1999)
6. Coxson, P.G.: Lumpability and observability of linear systems. *J. Math. Anal. Appl.* **99**, 435–446 (1984)
7. Derisavi, S., Hermanns, H., Sanders, W.H.: Optimal state-space lumping in Markov chains. *Inf. Process. Lett.* **87**(6), 309–315 (2003)
8. Gribaudo, M., Horváth, A.: Fluid stochastic Petri nets augmented with flush-out arcs: a transient analysis technique. *IEEE Trans. Softw. Eng.* **28**(10), 944–955 (2002)
9. Gribaudo, M., Sereno, M., Horváth, A., Bobbio, A.: Fluid stochastic Petri nets augmented with flush-out arcs: modelling and analysis. *Discret. Event Dyn. Syst.* **11**(1–2), 97–117 (2001)
10. Gribaudo, M., Telek, M.: Fluid models in performance analysis. In: Bernardo, M., Hillston, J. (eds.) SFM 2007. LNCS, vol. 4486, pp. 271–317. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-72522-0\\_7](https://doi.org/10.1007/978-3-540-72522-0_7)
11. Hillston, J.: *A Compositional Approach to Performance Modelling*. Cambridge University Press, Cambridge (1996)
12. Horton, G., Kulkarni, V.G., Nicol, D.M., Trivedi, K.S.: Fluid stochastic Petri nets: theory, applications, and solution techniques. *Eur. J. Oper. Res.* **105**(1), 184–201 (1998)
13. Iacobelli, G., Tribastone, M., Vandin, A.: Differential bisimulation for a Markovian process algebra. In: Italiano, G.F., Pighizzini, G., Sannella, D.T. (eds.) MFCS 2015. LNCS, vol. 9234, pp. 293–306. Springer, Heidelberg (2015). doi:[10.1007/978-3-662-48057-1\\_23](https://doi.org/10.1007/978-3-662-48057-1_23)

14. Jasak, H.: Error analysis and estimation for the finite volume method with applications to fluid flows. Ph.D. thesis, University of London, Department of Mechanical Engineering (1996)
15. Kemeny, J.G., Snell, J.L.: Finite Markov Chains. Springer, Heidelberg (1976)
16. Toth, J., Li, G., Rabitz, H., Tomlin, A.S.: The effect of lumping and expanding on kinetic differential equations. SIAM J. Appl. Math. **57**(6), 1531–1556 (1997)
17. Tschaikowski, M., Tribastone, M.: Exact fluid lumpability in Markovian process algebra. Theor. Comput. Sci. **538**, 140–166 (2014)
18. Valmari, A., Franceschinis, G.: Simple  $O(m \log n)$  time Markov chain lumping. In: Esparza, J., Majumdar, R. (eds.) TACAS 2010. LNCS, vol. 6015, pp. 38–52. Springer, Heidelberg (2010). doi:[10.1007/978-3-642-12002-2\\_4](https://doi.org/10.1007/978-3-642-12002-2_4)



Analytical and Stochastic Modelling Techniques and  
Applications

24th International Conference, ASMTA 2017,  
Newcastle-upon-Tyne, UK, July 10-11, 2017,  
Proceedings

Thomas, N.; Forshaw, M. (Eds.)

2017, X, 213 p. 65 illus., Softcover

ISBN: 978-3-319-61427-4