Bisimulation of Labeled State-to-Function Transition Systems of Stochastic Process Languages

D. Latella & M. Massink
CNR – Istituto di Scienza e Tecnologie dell'Informazione 'A. Faedo'

E.P. de Vink [*]
Technische Universiteit Eindhoven and Centrum Wiskunde Informatica

Abstract Labeled state-to-function transition systems, *FuTS* for short, admit multiple transition schemes from states to functions of finite support over general semirings. As such they constitute a convenient modeling instrument to deal with stochastic process languages. In this paper, the notion of bisimulation induced by a *FuTS* is addressed from a coalgebraic point of view. A correspondence result is proven stating that *FuTS*-bisimulation coincides with the behavioral equivalence of the associated functor. As generic examples, the concrete existing equivalences for the core of the stochastic process algebras *PEPA* and *IML* are related to the bisimulation of specific *FuTS*, providing via the correspondence result coalgebraic justification of the equivalences of these calculi.

1 Introduction

Process description languages equipped with formal operational semantics are successful formalisms for modeling concurrent systems and analyzing their behavior. Typically, the operational semantics is defined by means of a labeled transition system following the SOS approach. The states of the transition systems are just process terms, while the labels of the transitions between states represent the possible actions and interactions. Process description languages often come equipped with process equivalences, so that system models can be compared according to specific behavioral relations.

In the last couple of decades, process languages have been enriched with quantitative information. Among these quantitative extensions, those allowing a stochastic representation of time, usually referred to as stochastic process algebras, have received particular attention. The main aim has been the integration of qualitative descriptions and quantitative analysis in a single mathematical framework by building on the combination of labeled transition systems and continuous-time Markov chains. The latter being one of the most successful approaches to modeling and analyzing the performance of computer systems and networks. An overview on stochastic process algebras, equivalences and related analysis techniques can be found in [14, 1, 3], for example. A common feature of many stochastic process algebras is that actions are enriched with the rates of exponentially distributed random variables that characterize their duration. Although exploiting the same class of distributions, the models and the techniques underlying the definition of the calculi turn out to be significantly different in many respects. A prominent difference concerns the modeling of the race condition by means of the choice operator, and its relationship to the issue of transition multiplicity. In the quantitative setting, multiplicities can make a crucial distinction between processes that are qualitatively equivalent. Several significantly different approaches have been proposed for handling transition multiplicity. The proposals range from multi-relations [17, 13], to proved transition systems [23], to *LTS* with numbered transitions [14], to unique rate names [9], just to mention a few.

[*] Corresponding author, email `evink@win.tue.nl`.

U. Golas, T. Soboll (Eds.): Proceedings of ACCAT 2012
EPTCS 93, 2012, pp. 23–43, doi:10.4204/EPTCS.93.2

In [7], Latella, Massink et al. have proposed a variant of *LTS*, called Rate Transition Systems (*RTS*). In *LTS*, a transition is a triple (P, α, P') where P and α are the source state and the label of the transition, respectively, while P' is the target state reached from P via the transition. In *RTS*, a transition is a triple of the form $(P, \alpha, \mathcal{P})$. The first and second component are the source state and the label of the transition, as in *LTS*, while the third component $\mathcal{P}$ is a continuation function which associates a non-negative real value to each state P'. A non-zero value for the state P' represents the rate of the exponential distribution characterizing the time for the execution of the action represented by α, necessary to reach P' from P via the transition. If $\mathcal{P}$ maps P' to 0, then state P' is not reachable from P via the transition. The use of continuation functions provides a clean and simple solution to the transition multiplicity problem and make *RTS* particularly suited for stochastic process algebra semantics. In order to provide a uniform account of the many stochastic process algebras proposed in the literature, in previous joint work of the first two authors [8] Labelled State-to-Function Transition Systems (*FuTS*) have been introduced as a natural generalization of *RTS*. In *FuTS* the co-domain of the continuation functions are arbitrary semirings, rather than just the non-negative reals. This provides increased flexibility while preserving basic properties of primitive operations like sum and multiplication.

In this paper we present a coalgebraic treatment of *FuTS* that allow multiple state-to-function transition relations involving arbitrary semirings. Given label sets $\mathcal{L}_i$ and semirings $\mathcal{R}_i$, a *FuTS* takes the general format $\mathcal{S} = (S, \langle \rightarrowtail_i \rangle_{i=1}^{n})$ with transition relations $\rightarrowtail_i \subseteq S \times \mathcal{L}_i \times \mathcal{FS}(S, \mathcal{R}_i)$. Here, $\mathcal{FS}(S, \mathcal{R}_i)$ are the sets of functions from S to $\mathcal{R}_i$ of finite support, a subcollection of functions also occurring in other work combining coalgebra and quantitative modeling. We will associate to $\mathcal{S}$ the product of the functors $\mathcal{FS}(\cdot, \mathcal{R}_i)^{\mathcal{L}_i}$. For this to work, we need the transition relations $\rightarrowtail_i$ to be total and deterministic for the coalgebraic modeling as a function. Maybe surprisingly, this isn't a severe restriction at all in the presence of continuation functions: the zero-continuation $\lambda s'.0$ expresses that no *LTS*-transition exists from the state s to any state s'; if s allows a transition to some state s_1 as well as a state s_2, the continuation function will simply yield a non-zero value for s_1 and for s_2.

The notion of $\mathcal{S}$-bisimulation that arises from a *FuTS* $\mathcal{S}$ is reinterpreted coalgebraically as the behavioral equivalence of a functor that is induced by $\mathcal{S}$, along the lines sketched above. Behavioral equivalence rather than coalgebraic bisimulation is targeted, since, dependent on the semirings involved, weak pullbacks may not be preserved and the construction of a mediating morphism for a coalgebraic bisimulation from a concrete one may fail for degenerate denominators. However, following a familiar argument, we show that the functor associated with a *FuTS* does possess a final coalgebra and therefore has an associated notion of behavioral equivalence indeed. It is noted, in the presence of a final coalgebra for *FuTS* a more general definition of behavioral equivalence based on cospans coincides [20]. A correspondence result is proven in this paper that shows that the concrete bisimulation of a *FuTS*, coincides with behavioral equivalence of its functor. Pivotal for its proof is the absence of multiplicities in the *FuTS* treatment of quantities.

Using the bridge established by the correspondence result, we continue by showing for two well-known stochastic process algebras, viz. Hillston's *PEPA* [17] and Hermanns's *IML* [13], that the respective standard notion of strong equivalence and strong bisimulation coincides with behavioral equivalence of the associated *FuTS*. This constitutes the main contribution of the paper. *PEPA* stands out as one of the prominent Markovian process algebras, while *IML* specifically provides separate prefix constructions for actions and for delays. The equivalences of *PEPA* and of *IML* are compared with the bisimulations of the respective *FuTS* as given by an alternative operational semantics involving the state-to-function scheme. In passing, the multiplicities have to be dealt with. Appropriate lemmas are provided relating the relation-based cumulative treatment with *FuTS* to the multirelation-based explicit treatment of *PEPA* and *IML*.

Related work on coalgebra includes [28, 19, 26], papers that also cover measures and congruence formats, a topic not touched upon here. For the discrete parts, regarding the correspondence of bisimulations, our work aligns with the approach of the papers mentioned. In this paper the bialgebraic perspective of SOS and bisimulation [27] is left implicit. An interesting direction of research combining coalgebra and quantities studies various types of weighted automata, including linear weighted automata, and associated notions of bisimulation and languages, as well as algorithms for these notions [6, 25, 5]. In particular, building on a result on bounded functors [11], it is shown in [5] for a functor involving functions of finite support over a field that the final coalgebra exists. Below, we have followed the scheme of [5] to obtain such a result for a functor induced by a *FuTS*. The notions of equivalence addressed in this paper, as often in coalgebraic treatments of process relations, are all strong bisimilarities.

The present paper is organized as follows: Section 2 briefly discusses some material on semirings and coalgebras. Labeled state-to-function transition systems and *FuTS* as well as the associated notion of bisimulation are provided in Section 3. The coalgebraic counterparts of *FuTS* and *FuTS*-bisimulation are defined in Section 4, where we also establish the correspondence with behavioral equivalence of the final coalgebra. In Section 5 the standard equivalence of *PEPA* is identified with the bisimulation of a *FuTS* and, hence, with behavioral equivalence. In Section 6 the same is done for the language of *IMC* where actions and delays are present on equal footing. Section 7 wraps up and discusses directions of future research. An appendix provides the proofs of a number of lemmas.

2 Preliminaries

A tuple $\mathcal{R} = (R, +, 0, *, 1)$ is called a semiring, if $(R, +, 0)$ is a commutative monoid with neutral element 0, $(R, *, 1)$ is a monoid with neutral element 1, $*$ distributes over $+$, and $0 * r = r * 0 = 0$ for all $r \in R$. As examples of a semiring we will use are the booleans $\mathbb{B} = \{\texttt{false}, \texttt{true}\}$ with disjunction as sum and conjunction as multiplication, and the non-negative reals $\mathbb{R}_{\geqslant 0}$ with the standard operations. We will consider, for a semiring $\mathcal{R}$ and a function $\varphi : X \to \mathcal{R}$, countable sums $\sum_{x \in X'} \varphi(x)$ in $\mathcal{R}$, for $X' \subseteq X$. For such a sum to exist we require φ to be of finite support, i.e. the support set $spt(\varphi) = \{ x \in X \mid \varphi(x) \neq 0 \}$ is finite. Here, 0 is the neutral element of $\mathcal{R}$ with respect to $+$.

We use the notation $\mathcal{FS}(X,\mathcal{R})$ for the collection of all functions of finite support from the set X to the semiring $\mathcal{R}$. A construct $[\, x_1 \mapsto r_1, \ldots, x_n \mapsto r_n \,]$, with $x_i \in X$, $i = 1 \ldots n$ all distinct, $r_i \in \mathcal{R}$, $i = 1 \ldots n$, denotes the mapping that assigns r_i to x_i, $i = 1 \ldots n$, and assigns 0 to all $x \in X$ different from all x_i. In particular $[]$, or more precisely $[]_{\mathcal{R}}$, is the constant function $x \mapsto 0$ and $\mathcal{X}_x = [\, x \mapsto 1 \,]$ is the characteristic function on $\mathcal{R}$ for $x \in X$. For $\varphi \in \mathcal{FS}(X,\mathcal{R})$, we write $\oplus\varphi$ for the value $\sum_{x \in X} \varphi(x)$ in $\mathcal{R}$. For $\varphi, \psi \in \mathcal{FS}(X,\mathcal{R})$, the function $\varphi + \psi$ is the pointwise sum of φ and ψ, i.e. $(\varphi + \psi)(x) = \varphi(x) + \psi(x) \in \mathcal{R}$. Clearly, $\varphi + \psi$ is of finite support as φ and ψ are. Given an injective operation $| : X \times X \to X$, we define $\varphi \,|\, \psi : X \to \mathcal{R}$, by $(\varphi \,|\, \psi)(x) = \varphi(x_1) * \psi(x_2)$ if $x = x_1 \,|\, x_2$ for some $x_1, x_2 \in X$, and $(\varphi \,|\, \psi)(x) = 0$ otherwise. Again, $\varphi \,|\, \psi$ is of finite support as φ and ψ are. This is used in the setting of syntactic processes P that may have the form $P_1 \parallel_A P_2$ for two processes P_1 and P_2 and a syntactic operator $\parallel_A$.

Lemma 1. *Let X be a set, $\mathcal{R}$ a semiring, and $|$ an injective binary operation on X. For $\varphi, \psi \in \mathcal{FS}(X,\mathcal{R})$ it holds that $\oplus(\varphi + \psi) = \oplus\varphi + \oplus\psi$ and $\oplus(\varphi \,|\, \psi) = (\oplus\varphi) * (\oplus\psi)$.* □

We recall some basic definitions from coalgebra. See e.g. [24] for more details. For a functor $\mathcal{F} : \mathbf{Set} \to \mathbf{Set}$ on the category **Set** of sets and functions, a coalgebra of $\mathcal{F}$ is a set X together with a mapping $\alpha : X \to \mathcal{F}(X)$. A homomorphism between two $\mathcal{F}$-coalgebras (X,α) and (Y,β) is a function $f : X \to Y$ such that $\mathcal{F}(f) \circ \alpha = \beta \circ f$. An $\mathcal{F}$-coalgebra (Ω, ω) is called final, if there exists, for every $\mathcal{F}$-coalgebra

(X,α), a unique homomorphism $[\![\cdot]\!]^{\mathcal{F}}_X : (X,\alpha) \to (\Omega,\omega)$. Two elements x_1, x_2 of a coalgebra (X,α) are called behavioral equivalent with respect to $\mathcal{F}$ if $[\![x_1]\!]^{\mathcal{F}}_X = [\![x_2]\!]^{\mathcal{F}}_X$, notation $x_1 \approx_{\mathcal{F}} x_2$.

Using a characterization of [12], a functor $\mathcal{F}$ on **Set** is bounded, if there exist sets A and B and a surjective natural transformation $\eta : A \times (\cdot)^B \Rightarrow \mathcal{F}$. Here, $A \times (\cdot)$ is the functor that maps a set X to the Cartesian product $A \times X$ and maps a function $f : X \to Y$ to the mapping $A \times f : A \times X \to A \times Y$ with $(A \times f)(a,x) = (a, f(x))$, while $(\cdot)^B$ denotes the functor that maps a set X to the function space X^B of all functions from B to X and that maps a function $f : X \to Y$ to the mapping $f^B : X^B \to Y^B$ with $f^B(\varphi)(b) = f(\varphi(b))$. For bounded functors we have the following result, see [11] for a proof.

Theorem 2. *If a functor $\mathcal{F}$: **Set** $\to$ **Set** is bounded, then its final coalgebra exists.* □

A number of proofs of results on process languages $\mathcal{P}$ in this paper relies on so-called guarded recursion [2]. Typically, constants X are a syntactical ingredient in these languages. As usual, if $X := P$, i.e. the constant X is declared to have the process P as its body, we require P to be prefix-guarded. Thus, any occurrence of a constant in the body P is in the scope of a prefix-construct of the language. Guarded recursion assumes the existence of a function $c : \mathcal{P} \to \mathbb{N}$ such that $c(P_1 \bullet P_2) > \max\{c(P_1), c(P_2)\}$ for all syntactic operations $\bullet$ of $\mathcal{P}$, and moreover $c(X) > c(P)$ if $X := P$.

3 Labeled State-to-Function Transition Systems

The definition of a labeled state-to-function transition system, *FuTS* for short, involves a set of states S and one or more relations of states and functions from states into a semiring. For sums over arbitrary subsets of states to exist, the functions are assumed to be of finite support.

Definition 1. *A FuTS $\mathcal{S}$, in full a labeled state-to-function transition system, over a number of label sets $\mathcal{L}_i$ and semirings $\mathcal{R}_i$, $i = 1 \ldots n$, is a tuple $\mathcal{S} = (S, \langle \rightarrowtail_i \rangle_{i=1}^n)$ such that $\rightarrowtail_i \subseteq S \times \mathcal{L}_i \times \mathcal{FS}(S, \mathcal{R}_i)$, for $i = 1 \ldots n$.* •

As usual, we write $s \overset{\ell}{\rightarrowtail}_i v$ for $(s, \ell, v) \in \rightarrowtail_i$. For a *FuTS* $\mathcal{S} = (S, \langle \rightarrowtail_i \rangle_{i=1}^n)$ the set S is called the set of states. We refer to each $\rightarrowtail_i$ as a state-to-function transition relation of $\mathcal{S}$ or just as a transition relation of it. If for $\mathcal{S}$ we have that $n = 1$, i.e. there is only one state-to-function transition relation $\rightarrowtail$, then $\mathcal{S}$ is called simple. A *FuTS* $\mathcal{S}$ is called total and deterministic if for each transition relation $\rightarrowtail_i \subseteq S \times \mathcal{L}_i \times \mathcal{FS}(S, \mathcal{R}_i)$ involved and for all $s \in S$, $\ell \in \mathcal{L}_i$, we have $s \overset{\ell}{\rightarrowtail}_i v$ for exactly one $v \in \mathcal{FS}(S, \mathcal{R}_i)$. In such a situation, the zero-function $[\,]_{\mathcal{R}_i}$ plays a special role. A state-to-function transition $s \overset{\ell}{\rightarrowtail}_i [\,]_{\mathcal{R}_i}$ reflects the absence of a non-trivial transition $s \overset{\ell}{\rightarrowtail}_i v$ for $v \neq [\,]_{\mathcal{R}_i}$. In the context of *LTS* one says that s has no ℓ-transition. For the remainder of the paper, all *FuTS* we consider are assumed to be total and deterministic.[1]

Examples For the modeling of *CCS* processes, we choose a set of actions $\mathcal{A}$ as label set and the booleans $\mathbb{B}$ as semiring. Consider the two *CCS* processes $P = a.b.\mathbf{0} + a.c.\mathbf{0}$ and $Q = a.(b.\mathbf{0} + c.\mathbf{0})$, their representation as a *FuTS* is depicted in Figure 1. For process P we have $P \overset{a}{\rightarrowtail} [b.\mathbf{0} \mapsto \mathtt{true}, c.\mathbf{0} \mapsto \mathtt{true}]$, while the process Q we have $Q \overset{a}{\rightarrowtail} [b.\mathbf{0} + c.\mathbf{0} \mapsto \mathtt{true}]$. So, *FuTS* are able to represent branching.

As another example of a simple *FuTS*, Figure 1 displays a *FuTS* over the action set $\mathcal{A}$ and the semiring $\mathbb{R}_{\geqslant 0}$ of the non-negative real numbers. The functions v_0 to v_4 used in the example have the property that $\oplus v_i(s) = 1$, for $i = 0 \ldots 4$. Usually, such a *FuTS* over $\mathbb{R}_{\geqslant 0}$ is called a (reactive) probabilistic transition system.

[1] Definition 1 slightly differs in formulation from the one in [8].

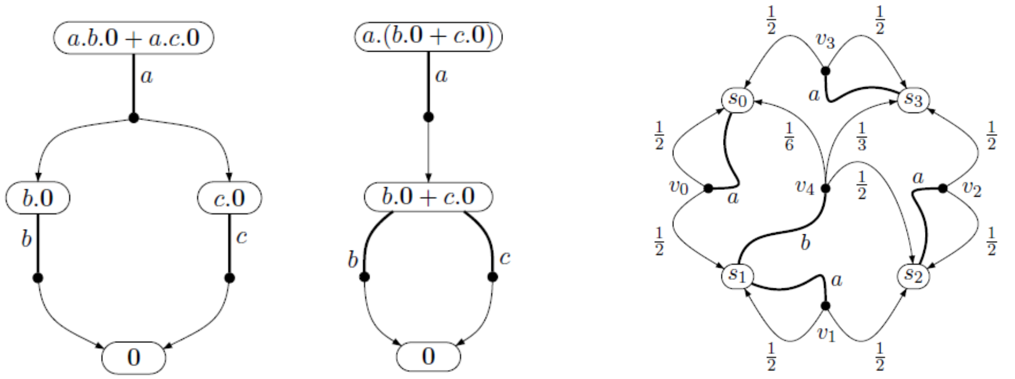

Figure 1: *FuTS* for two *CCS* processes and a probabilistic process.

In Section 6 we will provide semantics for the process language *IML* of interactive Markov chains [13, 16] using *FuTS*. Unlike many other stochastic process algebras, a single *IML* process can in general both perform action-based transitions and time-delays governed by exponential distributions.

It will be notationally convenient to consider a (total and deterministic) *FuTS* as a tuple $(S, \langle \theta_i \rangle_{i=1}^n)$ with transition functions $\theta_i : S \to \mathcal{L}_i \to \mathcal{FS}(S, \mathcal{R}_i)$, $i = 1 \ldots n$, rather than using the form $(S, \langle \rightarrowtail_i \rangle_{i=1}^n)$ that occurs more frequent for concrete examples in the literature. Alternatively, using disjoint unions, one could see a *FuTS* represented by a function $\theta' : S \to \bigoplus_{i=1}^n \mathcal{L}_i \to \bigoplus_{i=1}^n \mathcal{FS}(S, \mathcal{R}_i)$ satisfying the additional property that $\theta'(s)(\ell) \in \mathcal{FS}(S, \mathcal{R}_i)$ if $\ell \in \mathcal{L}_i$. As this fits less smoothly with the category-theoretical approach of Section 4, we stick to the former format. Note, an interpretation of a *FuTS* as a function $S \to \bigoplus_{i=1}^n (\mathcal{L}_i \to \mathcal{FS}(S, \mathcal{R}_i))$ does not suit our purposes as the *IML* example above illustrates.

We will use the notation with transition functions $\theta_i : S \to \mathcal{L}_i \to \mathcal{FS}(S, \mathcal{R}_i)$ to introduce the notion of bisimilarity for a *FuTS*.

Definition 2. *Let $\mathcal{S} = (S, \langle \theta_i \rangle_{i=1}^n)$ be a FuTS over the label sets $\mathcal{L}_i$ and semirings $\mathcal{R}_i$, $i = 1 \ldots n$. An equivalence relation $R \subseteq S \times S$ is called an $\mathcal{S}$-bisimulation if $R(s_1, s_2)$ implies*

$$\textstyle\sum_{t' \in [t]_R} \theta_i(s_1)(\ell)(t') = \sum_{t' \in [t]_R} \theta_i(s_2)(\ell)(t') \tag{1}$$

for all $t \in S$, $i = 1 \ldots n$ and $\ell \in \mathcal{L}_i$. Two elements $s_1, s_2 \in S$ are called $\mathcal{S}$-bisimilar if $R(s_1, s_2)$ for some $\mathcal{S}$-bisimulation R for $\mathcal{S}$. Notation $s_1 \sim_{\mathcal{S}} s_2$. •

We use the notation $[t]_R$ to denote the equivalence class of $t \in S$ with respect to R. Note that sums in equation (1) exist since the functions $\theta_i(s_j)(\ell) \in \mathcal{FS}(S, \mathcal{R}_i)$, $i = 1 \ldots n$, $j = 1, 2$, are of finite support. Hence, $\theta_i(s_j)(\ell)(t') = 0 \in \mathcal{R}_i$ for all but finitely many $t' \in [t]_R \subseteq S$.

For the combined *FuTS* of the two *CCS*-processes of Figure 1, the obvious equivalence relation relating $b.\mathbf{0}$ and $b.\mathbf{0} + c.\mathbf{0}$ is not a *FuTS*-bisimulation. Although $\sum_{t' \in [\mathbf{0}]_R} \theta(b.\mathbf{0})(b)(t') = \theta(b.\mathbf{0})(b)(\mathbf{0}) = \mathtt{true}$ and $\sum_{t' \in [\mathbf{0}]_R} \theta(b.\mathbf{0} + c.\mathbf{0})(b)(t') = \theta(b.\mathbf{0} + c.\mathbf{0})(b)(\mathbf{0}) = \mathtt{true}$, we have $\sum_{t' \in [\mathbf{0}]_R} \theta(b.\mathbf{0})(c)(t') = \mathtt{false}$, while $\sum_{t' \in [\mathbf{0}]_R} \theta(b.\mathbf{0} + c.\mathbf{0})(c)(t') = \mathtt{true}$, taking sums, i.e. disjunctions, in $\mathbb{B}$.

4 *FuTS* coalgebraically

In this section we will cast *FuTS* in the framework of coalgebras and prove a correspondence result of *FuTS*-bisimulation and behavioral equivalence for a suitable functor on **Set**.

Definition 3. *Let $\mathcal{L}$ be a set of labels and let $\mathcal{R}$ be a semiring. The functor $\mathcal{V}^{\mathcal{L}}_{\mathcal{R}} : \textbf{Set} \to \textbf{Set}$ assigns to a set X the function space $\mathcal{FS}(X,\mathcal{R})^{\mathcal{L}}$ of all functions $\varphi : \mathcal{L} \to \mathcal{FS}(X,\mathcal{R})$ and assigns to a function $f : X \to Y$ the mapping $\mathcal{V}^{\mathcal{L}}_{\mathcal{R}}(f) : \mathcal{FS}(X,\mathcal{R})^{\mathcal{L}} \to \mathcal{FS}(Y,\mathcal{R})^{\mathcal{L}}$ where*

$$\mathcal{V}^{\mathcal{L}}_{\mathcal{R}}(f)(\varphi)(\ell)(y) = \textstyle\sum_{x' \in f^{-1}(y)} \varphi(\ell)(x')$$

for all $\varphi \in \mathcal{FS}(X,\mathcal{R})^{\mathcal{L}}$, $\ell \in \mathcal{L}$ and $y \in Y$. •

Again we rely on $\varphi(\ell) \in \mathcal{FS}(X,\mathcal{R})$ having a finite support for the sum to exist and for $\mathcal{V}^{\mathcal{L}}_{\mathcal{R}}$ being well-defined. In fact, we have $spt(\mathcal{V}^{\mathcal{L}}_{\mathcal{R}}(f)(\varphi)(\ell)) = \{ f(x) \mid x \in spt(\varphi)(\ell) \}$.

As we aim to compare our notion of bisimulation for *FuTS* with behavioral equivalence for the functor $\mathcal{V}^{\mathcal{L}}_{\mathcal{R}}$, given a set of labels $\mathcal{L}$ and a semiring $\mathcal{R}$, we need to check that $\mathcal{V}^{\mathcal{L}}_{\mathcal{R}}$ possesses a final coalgebra. We follow the approach of [5].

Lemma 3. *Let $\mathcal{L}$ be a set of labels, $\mathcal{R}$ a semiring. Then the functor $\mathcal{V}^{\mathcal{L}}_{\mathcal{R}}$ on* ***Set*** *is bounded.* □

Working with total and deterministic *FuTS*, we can interpret a *FuTS* $\mathcal{S} = (S, \langle \theta_i \rangle_{i=1}^{n})$ over the label sets $\mathcal{L}_i$ and semirings $\mathcal{R}_i$, $i = 1 \ldots n$ as a product $\theta_1 \times \cdots \times \theta_n : S \to \prod_{i=1}^{n} (\mathcal{L}_i \to \mathcal{FS}(S,\mathcal{R}_i))$ of functions $\theta_i : S \to \mathcal{L}_i \to \mathcal{FS}(S,\mathcal{R}_i)$. To push this idea a bit further, we want to consider the *FuTS* $\mathcal{S} = (\mathcal{S}, \langle \theta_i \rangle_{i=1}^{n})$ as a coalgebra of a suitable product functor on **Set**.

Definition 4. *Let $\mathcal{S} = (S, \langle \theta_i \rangle_{i=1}^{n})$ be a FuTS over the label sets $\mathcal{L}_i$ and semirings $\mathcal{R}_i$, $i = 1..n$. The functor $\mathcal{V}_{\mathcal{S}}$ on* ***Set*** *is defined by $\mathcal{V}_{\mathcal{S}} = \prod_{i=1}^{n} \mathcal{V}^{\mathcal{L}_i}_{\mathcal{R}_i} = \prod_{i=1}^{n} \mathcal{FS}(\cdot,\mathcal{R}_i)^{\mathcal{L}_i}$.*

The point is, under conditions that are generally met, coalgebras come equipped with a natural notion of behavioral equivalence that can act as a reference for strong equivalences, in particular of bisimulation for *FuTS*. Below, see Theorem 5, we prove that $\mathcal{S}$-bisimilarity as given by Definition 2 coincides with behavioral equivalence for the functor $\mathcal{V}_{\mathcal{S}}$ as given by Definition 4, providing justification for the notion of equivalence defined on *FuTS*.

For the notion of behavioral equivalence for the functor $\mathcal{V}_{\mathcal{S}}$ obtained from $\mathcal{S}$ to be defined, we establish that it possesses a final coalgebra.

Theorem 4. *The functor $\mathcal{V}_{\mathcal{S}}$ has a final coalgebra.*

Proof. By Lemma 3 we have that each factor $\mathcal{V}^{\mathcal{L}_i}_{\mathcal{R}_i}$ of $\mathcal{V}_{\mathcal{S}}$ is bounded, and hence possesses a final coalgebra $\Omega_{\mathcal{V}^{\mathcal{L}_i}_{\mathcal{R}_i}}$ by Theorem 2. It follows that also $\mathcal{V}_{\mathcal{S}}$ has a final coalgebra $\Omega_{\mathcal{S}}$. Writing $[\![\cdot]\!]^{\mathcal{S}}_X$ for the final morphism of a $\mathcal{V}_{\mathcal{S}}$-coalgebra X into $\Omega_{\mathcal{S}}$, we have

$$\Omega_{\mathcal{S}} = \Omega_{\mathcal{V}^{\mathcal{L}_1}_{\mathcal{R}_1}} \times \cdots \times \Omega_{\mathcal{V}^{\mathcal{L}_n}_{\mathcal{R}_n}} \quad \text{and} \quad [\![\cdot]\!]^{\mathcal{S}}_X = [\![\cdot]\!]_X^{\mathcal{V}^{\mathcal{L}_1}_{\mathcal{R}_1}} \times \cdots \times [\![\cdot]\!]_X^{\mathcal{V}^{\mathcal{L}_n}_{\mathcal{R}_n}}$$

as can be straightforwardly shown. □

Since the functor $\mathcal{V}_{\mathcal{S}}$ of a *FuTS* has a final coalgebra, we can speak of the behavioral equivalence $\approx_{\mathcal{S}}$ induced by $\mathcal{V}_{\mathcal{S}}$. Next we establish, for a *FuTS* $\mathcal{S}$, the correspondence of $\mathcal{S}$-bisimulation $\sim_{\mathcal{S}}$ as given by Definition 2 and behavioral equivalence $\approx_{\mathcal{S}}$.

Theorem 5. *Let $\mathcal{S} = (S, \langle \theta_i \rangle_{i=1}^{n})$ be a FuTS over the label sets $\mathcal{L}_i$ and semirings $\mathcal{R}_i$, $i = 1 \ldots n$. Then $s_1 \sim_{\mathcal{S}} s_2 \Leftrightarrow s_1 \approx_{\mathcal{S}} s_2$, for all $s_1, s_2 \in S$.*

Proof. Let $s_1, s_2 \in S$. We first prove $s_1 \sim_{\mathcal{S}} s_2 \Rightarrow s_1 \approx_{\mathcal{S}} s_2$. So, assume $s_1 \sim_{\mathcal{S}} s_2$. Let $R \subseteq S \times S$ be an $\mathcal{S}$-bisimulation with $R(s_1, s_2)$. Put $\theta = \theta_1 \times \cdots \times \theta_n$. Note (S, θ) is a $\mathcal{V}_{\mathcal{S}}$-coalgebra. We turn the collection of equivalence classes S/R into a $\mathcal{V}_{\mathcal{S}}$-coalgebra $(S/R, \theta_R)$ by putting

$$\theta^i_R(\,[s]_R\,)(\ell)(\,[t]_R\,) = \textstyle\sum_{t' \in [t]_R} \theta_i(s)(\ell)(t') \quad \text{and} \quad \theta_R = \theta^1_R \times \cdots \times \theta^n_R$$

for $s, t \in S$, $\ell \in \mathcal{L}_i$, $i = 1 \ldots n$. This is well-defined since R is an $\mathcal{S}$-bisimulation: if $R(s, s')$ then we have $\sum_{t' \in [t]_R} \theta_i(s)(\ell)(t') = \sum_{t' \in [t]_R} \theta_i(s')(\ell)(t')$. The canonical mapping $\varepsilon_R : S \to S/R$ is a $\mathcal{V}_{\mathcal{S}}$-homomorphism: For $i = 1 \ldots n$, $\ell \in \mathcal{L}_i$ and $t \in S$, we have both

$$\mathcal{FS}(\varepsilon_R, \mathcal{R}_i)^{\mathcal{L}_i}(\theta_i(s))(\ell)([t]_R) = \textstyle\sum_{t' \in [t]_R} \theta_i(s)(\ell)(t') \quad \text{and} \quad \theta^i_R([s]_R)(\ell)([t]_R) = \textstyle\sum_{t' \in [t]_R} \theta_i(s)(\ell)(t')$$

Thus, $\mathcal{FS}(\varepsilon_R, \mathcal{R}_i)^{\mathcal{L}}_i \circ \theta_i = \theta^i_R \circ \varepsilon_R$. Since $\mathcal{V}_{\mathcal{S}}(\varepsilon_R) = \prod^n_{i=1} \mathcal{FS}(\varepsilon_R, \mathcal{R}_i)^{\mathcal{L}_i}$ it follows that ε_R is a $\mathcal{V}_{\mathcal{S}}$-homomorphism. Therefore, by uniqueness of a final morphism, we have $[\![\cdot]\!]^{\mathcal{S}}_S = [\![\cdot]\!]^{\mathcal{S}}_{S/R} \circ \varepsilon_R$. In particular, $[\![s_1]\!]^{\mathcal{S}}_S = [\![s_2]\!]^{\mathcal{S}}_S$ since $\varepsilon_R(s_1) = \varepsilon_R(s_2)$. Thus, $s_1 \approx_{\mathcal{S}} s_2$.

For the reverse, i.e. $s_1 \approx_{\mathcal{S}} s_2 \Rightarrow s_1 \sim_{\mathcal{S}} s_2$, assume $s_1 \approx_{\mathcal{S}} s_2$, i.e. $[\![s_1]\!]^{\mathcal{S}}_S = [\![s_2]\!]^{\mathcal{S}}_S$. Since the map $[\![\cdot]\!]^{\mathcal{S}}_S : (S, \theta) \to (\Omega_{\mathcal{S}}, \omega_{\mathcal{S}})$ is a $\mathcal{V}_{\mathcal{S}}$-homomorphism, the relation $R_{\mathcal{S}}$ with $R_{\mathcal{S}}(s', s'') \Leftrightarrow [\![s']\!]^{\mathcal{S}}_S = [\![s'']\!]^{\mathcal{S}}_S$ is an $\mathcal{S}$-bisimulation: Suppose $R_{\mathcal{S}}(s', s'')$, i.e. $s' \approx_{\mathcal{S}} s''$, for some $s', s'' \in \mathcal{S}$. Assume $\theta_{\Omega_{\mathcal{S}}} = \theta^1_{\Omega_{\mathcal{S}}} \times \cdots \times \theta^n_{\Omega_{\mathcal{S}}}$. Pick $1 \leqslant i \leqslant n$, $\ell \in \mathcal{L}_i$, $t \in S$. Put $[\![t]\!]^{\mathcal{S}}_S = w \in \Omega_{\mathcal{S}}$. Let $[t]_{\mathcal{S}}$ denote the equivalence class of t in $R_{\mathcal{S}}$.

$$\begin{array}{lll}
\multicolumn{3}{l}{\sum_{t' \in [t]_{\mathcal{S}}} \theta_i(s')(\ell)(t')} \\
\quad = & \sum_{t' \in ([\![\cdot]\!]^{\mathcal{S}}_S)^{-1}(w)} \theta_i(s')(\ell)(t') & \text{(by definition of } R_{\mathcal{S}} \text{ and } w\text{)} \\
\quad = & \theta^i_{\Omega_{\mathcal{S}}}([\![s']\!]^{\mathcal{S}}_S)(\ell)(w) & ([\![\cdot]\!]^{\mathcal{S}}_S \text{ is a } \mathcal{V}_{\mathcal{S}}\text{-homomorphism}) \\
\quad = & \theta^i_{\Omega_{\mathcal{S}}}([\![s'']\!]^{\mathcal{S}}_S)(\ell)(w) & (s' \approx_{\mathcal{S}} s'' \text{ by assumption}) \\
\quad = & \sum_{t' \in ([\![\cdot]\!]^{\mathcal{S}}_S)^{-1}(w)} \theta_i(s'')(\ell)(t') & ([\![\cdot]\!]^{\mathcal{S}}_S \text{ is a } \mathcal{V}_{\mathcal{S}}\text{-homomorphism}) \\
\quad = & \sum_{t' \in [t]_{\mathcal{S}}} \theta_i(s'')(\ell)(t') & \text{(by definition of } R_{\mathcal{S}} \text{ and } w\text{)}
\end{array}$$

Thus, if $R_{\mathcal{S}}(s', s'')$ then $\sum_{t' \in [t]_{\mathcal{S}}} \theta_i(s')(\ell)(t') = \sum_{t' \in [t]_{\mathcal{S}}} \theta_i(s'')(\ell)(t')$ for all $i = 1 \ldots n$, $t \in S$, $\ell \in \mathcal{L}_i$ and $R_{\mathcal{S}}$ is an $\mathcal{S}$-bisimulation. Since $[\![s_1]\!]^{\mathcal{S}}_S = [\![s_2]\!]^{\mathcal{S}}_S$, it follows that $R_{\mathcal{S}}(s_1, s_2)$. Thus $R_{\mathcal{S}}$ is an $\mathcal{S}$-bisimulation relating s_1 and s_2. Conclusion, it holds that $s_1 \sim_{\mathcal{S}} s_2$. □

5 *FuTS* Semantics of *PEPA*

Next we will consider a significant fragment of the process algebra *PEPA* [17], including the parallel operator implementing the scheme of so-called minimal apparent rates, and provide a *FuTS* semantics for it. We will show that *PEPA*'s notion of equivalence $=_{pepa}$, called strong equivalence in [17], fits with the bisimilarity $\sim_{pepa}$ as arising from the *FuTS* semantics.

Definition 5. *The set $\mathcal{P}_{\text{PEPA}}$ of PEPA processes is given by the BNF P ::= $\boldsymbol{nil} \mid (a, \lambda).P \mid P + P \mid P \parallel_A P \mid X$ where a ranges over the set of actions $\mathcal{A}$, λ over $\mathbb{R}_{>0}$, A over the set of finite subsets of $\mathcal{A}$, and X over the set of constants $\mathcal{X}$.* •

PEPA, like many other stochastic process algebras (e.g. [15, 4]), couples actions and rates. The prefix (a, λ) of the process $(a, \lambda).P$ expresses that the duration of the execution of the action $a \in \mathcal{A}$ is sampled from an exponential distribution of rate λ. The parallel composition $P \parallel_A Q$ of a process P and a process Q

$$\text{(NIL)}\ \frac{}{\mathbf{nil} \overset{\delta_a}{\rightarrowtail}_p []_{\mathbb{R}_{\geqslant 0}}} \qquad \text{(RAPF1)}\ \frac{}{(a,\lambda).P \overset{\delta_a}{\rightarrowtail}_p [P \mapsto \lambda]} \qquad \text{(RAPF2)}\ \frac{b \neq a}{(a,\lambda).P \overset{\delta_b}{\rightarrowtail}_p []_{\mathbb{R}_{\geqslant 0}}}$$

$$\text{(CHO)}\ \frac{P \overset{\delta_a}{\rightarrowtail}_p \mathcal{P} \quad Q \overset{\delta_a}{\rightarrowtail}_p \mathcal{Q}}{P + Q \overset{\delta_a}{\rightarrowtail}_p \mathcal{P} + \mathcal{Q}} \qquad \text{(CNS)}\ \frac{P \overset{\delta_a}{\rightarrowtail}_p \mathcal{P} \quad X := P}{X \overset{\delta_a}{\rightarrowtail}_p \mathcal{P}}$$

$$\text{(PAR1)}\ \frac{P \overset{\delta_a}{\rightarrowtail}_p \mathcal{P} \quad Q \overset{\delta_a}{\rightarrowtail}_p \mathcal{Q} \quad a \notin A}{P \parallel_A Q \overset{\delta_a}{\rightarrowtail}_p (\mathcal{P} \parallel_A \mathcal{X}_Q) + (\mathcal{X}_P \parallel_A \mathcal{Q})} \qquad \text{(PAR2)}\ \frac{P \overset{\delta_a}{\rightarrowtail}_p \mathcal{P} \quad Q \overset{\delta_a}{\rightarrowtail}_p \mathcal{Q} \quad a \in A}{P \parallel_A Q \overset{\delta_a}{\rightarrowtail}_p \mathit{arf}(\mathcal{P},\mathcal{Q}) \cdot (\mathcal{P} \parallel_A \mathcal{Q})}$$

Figure 2: *FuTS* semantics for *PEPA*.

for a set of actions $A \subseteq \mathcal{A}$ allows for the independent, asynchronous execution of actions of P and Q not occurring in the subset A, on the one hand, and requires the simultaneous, synchronized execution of P and Q for the actions occurring in A, on the other hand. The *FuTS*-semantics of the fragment of *PEPA* that we consider here, is given by the SOS of Figure 2, on which we comment below.

Characteristic for the *PEPA* language is the choice to model parallel composition, or cooperation in the terminology of *PEPA*, scaled by the minimum of the so-called apparent rates. By doing so, *PEPA*'s strong equivalence becomes a congruence [17]. Intuitively, the apparent rate $r_a(P)$ of an action a for a process P is the sum of the rates of all possible a-executions for P. When considering the CSP-style parallel composition $P \parallel_A Q$, with cooperation set A, an action a occurring in A has to be performed by both P and Q. The rate of such an execution is governed by the slowest, on average, of the two processes in this respect.[2] Thus $r_a(P \parallel_A Q)$ for $a \in A$ is the minimum $\min\{ r_a(P), r_a(Q) \}$. Now, if P schedules an execution of a with rate r_1 and Q schedules a transition of a with rate r_2, in the minimal apparent rate scheme the combined execution yields the action a with rate $r_1 \cdot r_2 \cdot \mathit{arf}(P,Q)$. Here, the 'syntactic' scaling factor $\mathit{arf}(P,Q)$, the apparent rate factor, is defined by

$$\mathit{arf}(P,Q) = \frac{\min\{ r_a(P), r_a(Q) \}}{r_a(P) \cdot r_a(Q)}$$

assuming $r_a(P), r_a(Q) > 0$, otherwise $\mathit{arf}(P,Q) = 0$. Thus, for $P \parallel_A Q$ the minimum $\min\{ r_a(P), r_a(Q) \}$ of the apparent rates is adjusted by the relative probabilities $r_1/r_a(P)$ and $r_2/r_a(Q)$ for executing a by P and Q, respectively. See [17, Definition 3.3.1] (or the appendix) for an explicit definition of the apparent rate r_a of a *PEPA*-process.

The *FuTS* we consider for the semantics of *PEPA* in Figure 2 involves a set of labels Δ defined by $\Delta = \{ \delta_a \mid a \in \mathcal{A} \}$. The symbol δ_a denotes the execution of the action a, with a duration that is still to be established. The underlying semiring for the simple *FuTS* for *PEPA* is the semiring $\mathbb{R}_{\geqslant 0}$ of non-negative reals.

Definition 6. *The FuTS $\mathcal{S}_{pepa} = (\mathcal{P}_{\mathit{PEPA}}, \rightarrowtail_p)$ over Δ and $\mathbb{R}_{\geqslant 0}$ has its transition relation given by the rules of Figure 2.* •

We discuss the rules of Figure 2. The *FuTS* semantics provides $\mathbf{nil} \overset{\delta_a}{\rightarrowtail}_p []_{\mathbb{R}_{\geqslant 0}}$, for every action a, with $[]_{\mathbb{R}_{\geqslant 0}}$ the 0-function $\lambda P.0$ of $\mathbb{R}_{\geqslant 0}$. However, the latter expresses $\theta_{pepa}(\mathbf{nil})(\delta_a)(P') = 0$ for every $a \in \mathcal{A}$

[2] One cannot take the slowest process per sample, because such an operation cannot be expressed as an exponential distribution in general.

and $P' \in \mathcal{P}_{PEPA}$, or, in standard terminology, **nil** has no transition. For the rated action prefix (a,λ) we distinguish two cases: (i) execution of the prefix in rule (RAPF1); (ii) no execution of the prefix in rule (RAPF2). In the case of rule (RAPF1) the label δ_a signifies that the transition involves the execution of the action a. The continuation $[\, P \mapsto \lambda \,]$ is the function that assigns the rate λ to the process P. All other processes are assigned 0, i.e. the zero-element of the semiring $\mathbb{R}_{\geqslant 0}$. In the second case, rule (RAPF2), for labels δ_b with $b \neq a$, we do have a transition, but it is a degenerate one. The two rules for the prefix, in particular having the 'null-continuation' rule (RAPF2), support the unified treatment of the choice operator in rule (CHO) and the parallel operator in rules (PAR1) and (PAR2).

Note the semantic sum of functions $\mathcal{P} + \mathcal{Q}$ replacing the syntactic sum in $P + Q$. The treatment of constants is as usual. Regarding the parallel operator $\|_A$, with respect to some subset of actions $A \subseteq \mathcal{A}$, the so-called cooperation set, there are again two rules. Now the distinction is between interleaving and synchronization. In the case of a label δ_a involving an action a not in the subset A, either the P-operand or the Q-operand of $P \|_A Q$ makes progress. For example, the effect of the pattern $\mathcal{P} \|_A \mathcal{X}_Q$ is that the value $\mathcal{P}(P') \cdot 1$ is assigned to a process $P' \|_A Q$, the value $\mathcal{P}(P') \cdot 0 = 0$ to a process $P' \|_A Q'$ for some $Q' \neq Q$, and the value 0 for a process not of the form $P' \|_A Q'$. Here, as in all other rules, the right-hand sides of the transitions only involve functions in $\mathcal{FS}(\mathcal{P}_{PEPA}, \mathbb{R}_{\geqslant 0})$ and operators on them.

For the synchronization case of the parallel construct, assuming $P \overset{\delta_a}{\rightarrowtail} \mathcal{P}$ and $Q \overset{\delta_a}{\rightarrowtail} \mathcal{Q}$, the 'semantic' scaling factor $\mathit{arf}(\mathcal{P},\mathcal{Q})$ is applied to $\mathcal{P} \|_A \mathcal{Q}$ (with $\|_A$ on $\mathcal{FS}(\mathcal{P}_{PEPA}, \mathbb{R}_{\geqslant 0})$ induced by $\|_A$ on $\mathcal{P}_{PEPA}$). This scaling factor, defined for functions in $\mathcal{FS}(\mathcal{P}_{PEPA}, \mathbb{R}_{\geqslant 0})$, is given by

$$\mathit{arf}(\mathcal{P}, \mathcal{Q}) = \frac{\min\{\oplus\mathcal{P}, \oplus\mathcal{Q}\}}{\oplus\mathcal{P} \cdot \oplus\mathcal{Q}}$$

provided $\oplus\mathcal{P}, \oplus\mathcal{Q} > 0$, and $\mathit{arf}(\mathcal{P}, \mathcal{Q}) = 0$ otherwise. This results for $\mathit{arf}(\mathcal{P},\mathcal{Q}) \cdot (\mathcal{P} \|_A \mathcal{Q})$, for a process $R = R_1 \|_A R_2$, in the value $\mathit{arf}(\mathcal{P}, \mathcal{Q}) \cdot (\mathcal{P} \|_A \mathcal{Q})(R_1 \|_A R_2) = \mathit{arf}(\mathcal{P}, \mathcal{Q}) \cdot \mathcal{P}(R_1) \cdot \mathcal{Q}(R_2)$.

The following lemma establishes the relationship between the 'syntactic' and 'semantic' apparent rate factors defined on processes and on continuation functions, respectively.

Lemma 6. *Let $P \in \mathcal{P}_{PEPA}$ and $a \in \mathcal{A}$. Suppose $P \overset{\delta_a}{\rightarrowtail}_p \mathcal{P}$. Then $r_a(P) = \oplus\mathcal{P}$.* □

The proof of the lemma is straightforward. It is also easy to prove, by guarded induction, that the *FuTS* $\mathcal{S}_{pepa}$ given by Definition 6 is total and deterministic. So, it is justified to write $\mathcal{S}_{pepa} = (\mathcal{P}_{PEPA}, \theta_{pepa})$. We use $\sim_{pepa}$ to denote the bisimilarity induced by $\mathcal{S}_{pepa}$.

Lemma 7. *The FuTS $\mathcal{S}_{pepa}$ is total and deterministic.* □

Example To illustrate the ease to deal with multiplicities in the *FuTS* semantics, consider the *PEPA* processes $P_1 = (a,\lambda).P$ and $P_2 = (a,\lambda).P + (a,\lambda).P$ for some $P \in \mathcal{P}_{PEPA}$. We have $P_1 \overset{\delta_a}{\rightarrowtail}_p [\, P \mapsto \lambda \,]$ by rule (RAPF1), but $P_2 \overset{\delta_a}{\rightarrowtail}_p [\, P \mapsto 2\lambda \,]$ by rule (RAPF1) and rule (CHO). The latter makes us to compute $[\, P \mapsto \lambda \,] + [\, P \mapsto \lambda \,]$, which equals $[\, P \mapsto 2\lambda \,]$. Thus, in particular we have $P_1 \nsim_{pepa} P_2$. Intuitively it is clear that, in general we cannot have $P + P \sim P$ for any reasonable quantitative process equivalence $\sim$ in the Markovian setting. Having twice as many a-labelled transitions, the average number for $(a,\lambda).P + (a,\lambda).P$ of executing the action a per time unit is double the average of executing a for $(a,\lambda).P$.

The standard operational semantics of *PEPA* [17, 18] is given in Figure 3. The transition relation $\rightarrow \subseteq \mathcal{P}_{PEPA} \times (\mathcal{A} \times \mathbb{R}_{>0}) \times \mathcal{P}_{PEPA}$ is the least relation satisfying the rules. For a proper treatment of the rates, the transition relation is considered as a multi-transition system, where also the number of possible

$$\text{(RAPF)}\ \frac{}{(a,\lambda).P \xrightarrow{a,\lambda} P} \qquad \text{(CHO1)}\ \frac{P \xrightarrow{a,\lambda} P'}{P + Q \xrightarrow{a,\lambda} P'} \qquad \text{(CHO2)}\ \frac{Q \xrightarrow{a,\lambda} Q'}{P + Q \xrightarrow{a,\lambda} P'}$$

$$\text{(PAR1a)}\ \frac{P \xrightarrow{a,\lambda} P' \quad a \notin A}{P \parallel_A Q \xrightarrow{a,\lambda} P' \parallel_A Q} \qquad \text{(PAR1b)}\ \frac{Q \xrightarrow{a,\lambda} Q' \quad a \notin A}{P \parallel_A Q \xrightarrow{a,\lambda} P \parallel_A Q'} \qquad \text{(CNS)}\ \frac{P \xrightarrow{a,\lambda} P' \quad X := P}{X \xrightarrow{a,\lambda} P'}$$

$$\text{(PAR2)}\ \frac{P \xrightarrow{a,\lambda_1} P' \quad Q \xrightarrow{a,\lambda_2} Q' \quad a \in A}{P \parallel_A Q \xrightarrow{a,\lambda} P' \parallel_A Q'} \quad \lambda = \mathit{arf}(P,Q)\cdot\lambda_1\cdot\lambda_2$$

Figure 3: Standard semantics for *PEPA*.

derivations of a transition $P \xrightarrow{a,\lambda} P'$ matters. We stress that such bookkeeping is not needed in the *FuTS*-approach at all. In rule (PAR2) we use the 'syntactic' apparent rate factor for *PEPA* processes.

The so-called total conditional transition rate $q[P,C,a]$ of a *PEPA*-process [17, 18] for a subset of processes $C \subseteq \mathcal{P}_{\mathit{PEPA}}$ and $a \in \mathcal{A}$ is given by $q[P,C,a] = \sum_{Q \in C} \sum \{\!|\, \lambda \mid P \xrightarrow{a,\lambda} Q \,|\!\}$. Here, $\{\!|\, P \xrightarrow{a,\lambda} Q \,|\!\}$ is the multiset of transitions $P \xrightarrow{a,\lambda} Q$ and $\{\!|\, \lambda \mid P \xrightarrow{a,\lambda} Q \,|\!\}$ is the multiset of all λ's involved. The multiplicity of $P \xrightarrow{a,\lambda} Q$ is the number of different ways the transition can be derived using the rules of Figure 3. We are now ready to define *PEPA*'s notion of strong equivalence [17, 18].

Definition 7. *An equivalence relation $R \subseteq \mathcal{P}_{\mathit{PEPA}} \times \mathcal{P}_{\mathit{PEPA}}$ is called a strong equivalence if $q[P_1,[Q]_R,a] = q[P_2,[Q]_R,a]$ for all $P_1,P_2 \in \mathcal{P}_{\mathit{PEPA}}$ such that $R(P_1,P_2)$, all $Q \in \mathcal{P}_{\mathit{PEPA}}$ and all $a \in \mathcal{A}$. Two processes $P_1,P_2 \in \mathcal{P}_{\mathit{PEPA}}$ are strongly equivalent if $R(P_1,P_2)$ for a strong equivalence R, notation $P_1 =_{\mathit{pepa}} P_2$.* •

The next lemma couples, for a *PEPA*-process P, an action a and a function $\mathcal{P} \in \mathcal{FS}(\mathcal{P}_{\mathit{PEPA}}, \mathbb{R}_{\geqslant 0})$, the evaluation $\mathcal{P}(P')$ with respect to the *FuTS*-semantics to the cumulative rate for P of reaching P' by a transition involving the label a in the standard operational semantics.

Lemma 8. *Let $P \in \mathcal{P}_{\mathit{PEPA}}$ and $a \in \mathcal{A}$. Suppose $P \overset{\delta_a}{\rightarrowtail} \mathcal{P}$. Then it holds that $\mathcal{P}(P') = \sum \{\!|\, \lambda \mid P \xrightarrow{a,\lambda} P' \,|\!\}$ for all $P' \in \mathcal{P}_{\mathit{PEPA}}$.* □

With the lemma in place we can prove the following correspondence result for $\mathcal{S}_{\mathit{pepa}}$-bisimilarity with respect to the *FuTS* for *PEPA* of Definition 6 and strong equivalence as given by Definition 7.

Theorem 9. *For any two PEPA-processes $P_1,P_2 \in \mathcal{P}_{\mathit{PEPA}}$ it holds that $P_1 \sim_{\mathit{pepa}} P_2$ iff $P_1 =_{\mathit{pepa}} P_2$.*

Proof. Let R be an equivalence relation on $\mathcal{P}_{\mathit{PEPA}}$. Choose $P,Q \in \mathcal{P}_{\mathit{PEPA}}$ and $a \in \mathcal{A}$. Suppose $P \overset{\delta_a}{\rightarrowtail}_p \mathcal{P}$. Thus $\theta_{\mathit{pepa}}(P)(\delta_a) = \mathcal{P}$. We have

$$\begin{array}{rcll} q[P,[Q]_R,a] & = & \sum_{Q' \in [Q]_R} \sum \{\!|\, \lambda \mid P \xrightarrow{a,\lambda} Q' \,|\!\} & \text{(by definition } q[P,[Q]_R,a]\text{)} \\ & = & \sum_{Q' \in [Q]_R} \mathcal{P}(Q') & \text{(by Lemma 8)} \\ & = & \sum_{Q' \in [Q]_R} \theta_{\mathit{pepa}}(P)(a)(Q') & \text{(by definition } \theta_{\mathit{pepa}}\text{)} \end{array}$$

Therefore, for *PEPA*-processes P_1 and P_2 it holds that $q[P_1,[Q]_R,a] = q[P_2,[Q]_R,a]$ for all $Q \in \mathcal{P}_{\mathit{PEPA}}$, $a \in \mathcal{A}$ iff $\sum_{Q' \in [Q]_R} \theta_{\mathit{pepa}}(P_1)(a)(Q') = \sum_{Q' \in [Q]_R} \theta_{\mathit{pepa}}(P_2)(a)(Q')$ for all $Q \in \mathcal{P}_{\mathit{PEPA}}$, $a \in \mathcal{A}$. Thus, the equivalence relation R is a strong equivalence iff R is an $\mathcal{S}_{\mathit{pepa}}$-bisimulation, from which the theorem follows. □

$$\text{(NIL1)}\ \frac{a \in \mathcal{A}}{\mathbf{nil} \overset{a}{\rightarrowtail}_1 []_{\mathbb{B}}} \qquad \text{(NIL2)}\ \frac{}{\mathbf{nil} \overset{\delta}{\rightarrowtail}_2 []_{\mathbb{R}_{\geqslant 0}}} \qquad \text{(APF3)}\ \frac{}{a.P \overset{\delta}{\rightarrowtail}_2 []_{\mathbb{R}_{\geqslant 0}}}$$

$$\text{(APF1)}\ \frac{}{a.P \overset{a}{\rightarrowtail}_1 [P \mapsto \texttt{true}]} \qquad \text{(APF2)}\ \frac{b \neq a}{a.P \overset{b}{\rightarrowtail}_1 []_{\mathbb{B}}} \qquad \text{(RPF1)}\ \frac{a \in \mathcal{A}}{\lambda.P \overset{a}{\rightarrowtail}_1 []_{\mathbb{B}}} \qquad \text{(RPF2)}\ \frac{}{\lambda.P \overset{\delta}{\rightarrowtail}_2 [P \mapsto \lambda]}$$

$$\text{(PAR1)}\ \frac{P \overset{\alpha}{\rightarrowtail}_i \mathcal{P} \quad Q \overset{\alpha}{\rightarrowtail}_i \mathcal{Q} \quad \alpha \notin A}{P \parallel_A Q \overset{\alpha}{\rightarrowtail}_i (\mathcal{P} \parallel_A \mathcal{X}^i_Q) + (\mathcal{X}^i_P \parallel_A \mathcal{Q})} (i = 1,2) \qquad \text{(PAR2)}\ \frac{P \overset{a}{\rightarrowtail}_1 \mathcal{P} \quad Q \overset{a}{\rightarrowtail}_1 \mathcal{Q} \quad a \in A}{P \parallel_A Q \overset{a}{\rightarrowtail}_1 \mathcal{P} \parallel_A \mathcal{Q}}$$

$$\text{(CHO)}\ \frac{P \overset{\alpha}{\rightarrowtail}_i \mathcal{P} \quad Q \overset{\alpha}{\rightarrowtail}_i \mathcal{Q}}{P + Q \overset{\alpha}{\rightarrowtail}_i \mathcal{P} + \mathcal{Q}} (i = 1,2) \qquad \text{(CON)}\ \frac{P \overset{\alpha}{\rightarrowtail}_i \mathcal{P} \quad X := P}{X \overset{\alpha}{\rightarrowtail}_i \mathcal{P}} (i = 1,2)$$

Figure 4: *FuTS* semantics for *IML*.

In view of our general correspondence result Theorem 5, the above theorem shows that *PEPA*'s strong equivalence $=_{pepa}$ is a behavioral equivalence, viz. behavioral equivalence $\approx_{pepa}$ with respect to the functor of $\mathcal{S}_{pepa}$, and that its standard, *FuTS* and coalgebraic semantics coincide.

6 *FuTS* Semantics of *IML*

In this section we provide a *FuTS* semantics for a relevant part of the language of *IMC* [13]. *IMC*, Interactive Markov Chains, are automata that combine two types of transitions: interactive transitions that involve the execution of actions and Markovian transitions that represent the progress of time governed by exponential distribution. As a consequence, *IMC* embody both non-deterministic and stochastic behaviour. System analysis using *IMC* proves to be a powerful approach because of the orthogonality of qualitative and quantitative dynamics, their logical underpinning and tool support. A number of equivalences, both strong and weak, are available for *IMC* [10]. In our treatment here, dealing with a fragment we call *IML*, we do not deal with internal τ-steps and focus on strong bisimulation.

Definition 8. *The set $\mathcal{P}_{IML}$ of IML processes is given by the BNF $P ::= \mathbf{nil} \mid a.P \mid \lambda.P \mid P+P \mid P \parallel_A P \mid X$ where a ranges over the set of actions $\mathcal{A}$, λ over $\mathbb{R}_{>0}$, A over the set of finite subsets of $\mathcal{A}$ and X over the set of constants $\mathcal{X}$.* •

In *IML* there are separate prefix constructions for actions $a.P$ and for time-delays $\lambda.P$. No restriction is imposed on the alternative and parallel composition of processes. For example, we have the process $P = a.\lambda.\mathbf{nil} + \mu.b.\mathbf{nil}$ in *IML*. It should be noted that for *IMC* actions are considered to take no time.

Definition 9. *The formal semantics of $\mathcal{P}_{IML}$ is given by the FuTS $\mathcal{S}_{iml} = (\mathcal{P}_{IML}, \rightarrowtail_1, \rightarrowtail_2)$ over the label sets $\mathcal{A}$ and $\Delta = \{\delta\}$ and the semirings $\mathbb{B}$ and $\mathbb{R}_{\geqslant 0}$ with transition relations $\rightarrowtail_1 \subseteq \mathcal{P}_{IML} \times \mathcal{A} \times \mathcal{FS}(\mathcal{P}_{IML}, \mathbb{B})$ and $\rightarrowtail_2 \subseteq \mathcal{P}_{IML} \times \Delta \times \mathcal{FS}(\mathcal{P}_{IML}, \mathbb{R}_{\geqslant 0})$ defined as the least relations satisfying the rules of Figure 4.* •

To accommodate for action-based and delay-related transitions, the *FuTS* $\mathcal{S}_{iml}$ is non-simple, having the two transition-to-function relations $\rightarrowtail_1$ and $\rightarrowtail_2$. Actions $a \in \mathcal{A}$ decorate $\rightarrowtail_1$, the special symbol δ decorates $\rightarrowtail_2$. Note rule (APF3) and rule (RPF1) that involve the null-functions of $\mathbb{R}_{\geqslant 0}$ and of $\mathbb{B}$, respectively, to express that a process $a.P$ does not trigger a delay and a process $\lambda.P$ does not execute an action. For the

$$\text{(APF)}\ \frac{}{a.P \xrightarrow{a} P} \qquad \text{(CHO1)}\ \frac{P \xrightarrow{a} R}{P + Q \xrightarrow{a} R} \qquad \text{(CHO2)}\ \frac{Q \xrightarrow{a} R}{P + Q \xrightarrow{a} R} \qquad \text{(CON1)}\ \frac{P \xrightarrow{a} Q \quad X := P}{X \xrightarrow{a} Q}$$

$$\text{(PAR1a)}\ \frac{P \xrightarrow{a} P' \quad a \notin A}{P \parallel_A Q \xrightarrow{a} ,P' \parallel_A Q} \qquad \text{(PAR1b)}\ \frac{Q \xrightarrow{a} Q' \quad a \notin A}{P \parallel_A Q \xrightarrow{a} P \parallel_A Q'} \qquad \text{(PAR2)}\ \frac{P \xrightarrow{a} P' \quad Q \xrightarrow{a} Q' \quad a \in A}{P \parallel_A Q \xrightarrow{a} P' \parallel_A Q'}$$

$$\text{(RPF)}\ \frac{}{\lambda.P \overset{\lambda}{\dashrightarrow} P} \qquad \text{(CHO3)}\ \frac{P \overset{\lambda}{\dashrightarrow} R}{P + Q \overset{\lambda}{\dashrightarrow} R} \qquad \text{(CHO4)}\ \frac{Q \overset{\lambda}{\dashrightarrow} R}{P + Q \overset{\lambda}{\dashrightarrow} R} \qquad \text{(CON2)}\ \frac{P \overset{\lambda}{\dashrightarrow} Q \quad X := P}{X \overset{\lambda}{\dashrightarrow} Q}$$

$$\text{(PAR1c)}\ \frac{P \overset{\lambda}{\dashrightarrow} P'}{P \parallel_A Q \overset{\lambda}{\dashrightarrow} P' \parallel_A Q} \qquad \text{(PAR1d)}\ \frac{Q \overset{\lambda}{\dashrightarrow} Q'}{P \parallel_A Q \overset{\lambda}{\dashrightarrow} P \parallel_A Q'}$$

Figure 5: Standard SOS rules for *IML*.

parallel construct $\parallel_A$, interleaving applies both for non-synchronized actions $a \notin A$ as well as for delays (but not mixed). Therefore, rule (PAR1) pertains to both $\rightarrowtail_1$ and $\rightarrowtail_2$, with α ranging over $\mathcal{A} \cup \Delta$. The same holds for non-deterministic choice, rule (CHO), and constants, rule (CON). Finally, *IML* does not provide synchronization of delays in the parallel construct. Rule (PAR2) only concerns the transition relation $\rightarrowtail_2$. In rule (PAR1), for clarity, we decorated the characteristic functions, writing $\mathcal{X}^i_P$, for $i = 1,2$, for $\mathcal{X}_P = [\,P \mapsto \texttt{true}\,]$ in $\mathcal{FS}(\mathcal{P}_{IML},\mathbb{B}\,)$ and $\mathcal{X}_P = [\,P \mapsto 1\,]$ in $\mathcal{FS}(\mathcal{P}_{IML},\mathbb{R}_{\geqslant 0}\,)$.

Example Assume $X := a.\lambda.b.X$ and $Y := a.\mu.b.Y$. Put $A = \{a,b\}$. Then we have

$$\begin{array}{rll} X \parallel_A Y & \overset{a}{\rightarrowtail}_1 [\,\lambda.b.X \parallel_A \mu.b.Y \mapsto \texttt{true}\,] & \qquad \lambda.b.X \parallel_A \mu.b.Y \overset{\delta}{\rightarrowtail}_2 [\,b.X \parallel_A \mu.b.Y \mapsto \lambda,\ \lambda.b.X \parallel_A b.Y \mapsto \mu\,] \\ b.X \parallel_A b.Y & \overset{b}{\rightarrowtail}_1 [\quad X \parallel_A Y \quad \mapsto \texttt{true}\,] & \qquad b.X \parallel_A \mu.b.Y \overset{\delta}{\rightarrowtail}_2 [\,b.X \parallel_A b.Y \quad \mapsto \mu\,] \\ & & \qquad \lambda.b.X \parallel_A b.Y \quad \overset{\delta}{\rightarrowtail}_2 [\,b.X \parallel_A b.Y \quad \mapsto \lambda\,] \end{array}$$

It is not difficult to verify that $\mathcal{S}_{iml}$ is a total and deterministic *FuTS*. Below we use $\mathcal{S}_{iml} = (\mathcal{P}_{IML}, \theta_1, \theta_2)$ and write $\sim_{iml}$ for the associated bisimilarity.

Lemma 10. *The FuTS $\mathcal{S}_{iml}$ is total and deterministic.* □

The standard SOS semantics of *IML* [13] is given in Figure 5 involving the transition relations

$$\rightarrow\ \subseteq \mathcal{P}_{IML} \times \mathcal{A} \times \mathcal{P}_{IML} \qquad \text{and} \qquad \dashrightarrow\ \subseteq \mathcal{P}_{IML} \times \mathbb{R}_{>0} \times \mathcal{P}_{IML}$$

Below we will use the functions $\mathbf{T}$ and $\mathbf{R}$ based on $\rightarrow$ and $\dashrightarrow$, cf. [16]. We have $\mathbf{T}\colon \mathcal{P}_{IML} \times \mathcal{A} \times \mathbf{2}^{\mathcal{P}_{IML}} \to \mathbb{B}$ given by $\mathbf{T}(P,a,C) = \texttt{true}$ if the set $\{\,P' \in C \mid P \xrightarrow{a} P'\,\}$ is non-empty, for all $P \in \mathcal{P}_{IML}$, $a \in \mathcal{A}$ and any subset $C \subseteq \mathcal{P}_{IML}$. For $\mathbf{R}\colon \mathcal{P}_{IML} \times \mathcal{P}_{IML} \to \mathbb{R}_{\geqslant 0}$ we put $\mathbf{R}(P,P') = \sum \{\!\!\{\, \lambda \mid P \overset{\lambda}{\dashrightarrow} P' \,\}\!\!\}$. Here, as common for probabilistic and stochastic process algebras, the comprehension is over the multiset of transitions leading from P to P' with label λ. We extend $\mathbf{R}$ to $\mathcal{P}_{IML} \times \mathbf{2}^{\mathcal{P}_{IML}}$ by $\mathbf{R}(P,C) = \sum_{P' \in C} \sum \{\!\!\{\, \lambda \mid P \overset{\lambda}{\dashrightarrow} P' \,\}\!\!\}$.

For *IML* we have the following notion of strong bisimulation [13, 16] that we will compare with the notion of bisimulation associated with the *FuTS* $\mathcal{S}_{iml}$.

Definition 10. *An equivalence relation $R \subseteq \mathcal{P}_{\mathrm{IML}} \times \mathcal{P}_{\mathrm{IML}}$ is called a strong bisimulation for IML if, for all $P_1, P_2 \in \mathcal{P}_{\mathrm{IML}}$ it holds that*

- *for all $a \in \mathcal{A}$ and $Q \in \mathcal{P}_{\mathrm{IML}}$: $\boldsymbol{T}(P_1, a, [Q]_R) \iff \boldsymbol{T}(P_2, a, [Q]_R)$*
- *for all $Q \in \mathcal{P}_{\mathrm{IML}}$: $\boldsymbol{R}(P_1, [Q]_R) = \boldsymbol{R}(P_2, [Q]_R)$.*

for all $P_1, P_2 \in \mathcal{P}_{\mathrm{IML}}$ such that $R(P_1, P_2)$. Two processes $P_1, P_2 \in \mathcal{P}_{\mathrm{IML}}$ are called strongly bisimilar if $R(P_1, P_2)$ for a strong bisimulation R for IML, notation $P_1 =_{iml} P_2$. •

To establish the correspondence of *FuTS* bisimilarity $\sim_{iml}$ for $\mathcal{S}_{iml}$ of Definition 9 and strong bisimilarity $=_{iml}$ for *IML*, we need to connect the state-to-function relation $\rightarrowtail_1$ and the transition relation $\rightarrow$ as well as the state-to-function relation $\rightarrowtail_2$ and the transition relation $\dashrightarrow$.

Lemma 11.

(a) *Let $P \in \mathcal{P}_{\mathrm{IML}}$ and $a \in \mathcal{A}$. If $P \overset{a}{\rightarrowtail}_1 \mathcal{P}$ then $P \xrightarrow{a} P' \iff \mathcal{P}(P') = \mathtt{true}$.*

(b) *Let $P \in \mathcal{P}_{\mathrm{IML}}$. If $P \overset{\delta}{\rightarrowtail}_2 \mathcal{P}$ then $\sum \{\!| \lambda \mid P \overset{\lambda}{\dashrightarrow} P' |\!\} = \mathcal{P}(P')$.* □

We are now in a position to relate *FuTS* bisimulation and standard strong bisimulation for *IML*.

Theorem 12. *For any two processes $P_1, P_2 \in \mathcal{P}_{\mathrm{IML}}$ it holds that $P_1 \sim_{iml} P_2$ iff $P_1 =_{iml} P_2$.*

Proof. Let R be an equivalence relation on $\mathcal{P}_{\mathrm{IML}}$. Pick $P \in \mathcal{P}_{\mathrm{IML}}$, $a \in \mathcal{A}$ and choose any $Q \in \mathcal{P}_{\mathrm{IML}}$. Suppose $P \overset{a}{\rightarrowtail} \mathcal{P}$. Thus $\theta_1(P)(a) = \mathcal{P}$. Then we have

$$\begin{array}{rcll} \mathbf{T}(P, a, [Q]_R) & \Leftrightarrow & \exists Q' \in [Q]_R \colon P \xrightarrow{a} Q' & \text{(by definition of } \mathbf{T}) \\ & \Leftrightarrow & \exists Q' \in [Q]_R \colon \mathcal{P}(Q') = \mathtt{true} & \text{(by Lemma 11a)} \\ & \Leftrightarrow & \sum_{Q' \in [Q]_R} \theta_1(P)(a)(Q) = \mathtt{true} & \text{(by definition of } \theta_1) \end{array}$$

Note, summation in $\mathbb{B}$ is disjunction. Likewise, on the quantitative side, we have

$$\begin{array}{rcll} \mathbf{R}(P, [Q]_R) & = & \sum_{Q' \in [Q]_R} \sum \{\!| \lambda \mid P \overset{\lambda}{\dashrightarrow} Q' |\!\} & \text{(by definition of } \mathbf{R}) \\ & = & \sum_{Q' \in [Q]_R} \mathcal{P}(Q') & \text{(by Lemma 11b)} \\ & = & \sum_{Q' \in [Q]_R} \theta_2(P)(\delta)(Q) & \text{(by definition of } \theta_2) \end{array}$$

Combining the equations, we conclude that a strong bisimulation for *IML* is also a bisimulation for the *FuTS* $\mathcal{S}_{iml}$, and vice versa. From this the theorem follows. □

Again, as a corollary of the theorem above, we have for *IML* that its notion of strong bisimulation is coalgebraically underpinned, as it coincides, calling to Theorem 5 once more, with behavioral equivalence of the functor $\mathcal{V}_{iml}$ induced by the *FuTS* $\mathcal{S}_{iml}$. As a consequence, the standard, *FuTS* and coalgebraic semantics for *IML* are all equal.

7 Concluding remarks

Total and deterministic labeled state-to-function transition systems, *FuTS*, are a convenient instrument to express the operational semantics of both qualitative and quantitative process languages. In this paper we have introduced the notion of bisimulation that arises from a *FuTS*, possibly involving multiple transition relations. A correspondence result, Theorem 5, relates the bisimulation of a *FuTS* $\mathcal{S}$ to behavioral equivalence of the functor $\mathcal{V}_\mathcal{S}$ that arises from the *FuTS* $\mathcal{S}$ too. For two prototypical stochastic process languages based on *PEPA* and on *IMC* we have shown that the notion of stochastic bisimulation associated with these calculi, coincides with the notion of bisimulation of the corresponding *FuTS*. Using these *FuTS* as a stepping stone, the correspondence result bridges between the concrete notion of bisimulation for *PEPA* and *IMC*, and the coalgebraic notion of behavioral equivalence. Hence, from this perspective, the concrete notions are seen as the natural strong equivalence to consider.

It is shown in [5], in the context of weighted automata, that in general the type of functors $\mathcal{FS}(\cdot,\mathcal{R})$ may not preserve weak pullbacks and, therefore, the notions of coalgebraic bisimulation and of behavioral equivalence may not coincide. Essential for the construction in their setting is the fact that the sum of non-zero weights may add to weight 0. The same phenomenon prevents a general proof, along the lines of [28], for coalgebraic bisimulation and *FuTS* bisimulation to coincide. In the construction of a mediating morphism, going from *FuTS* bisimulation to coalgebraic bisimulation a denominator may be zero, hence a division undefined, in case the sum over an equivalence class cancels out. In the concrete case for [19], although no detailed proof is provided there, this will not happen with $\mathbb{R}_{\geqslant 0}$ as underlying semiring. We expect that for semirings enjoying the property that for a sum $x = x_1 + \cdots + x_n$ it holds that $x = 0$ iff $x_i = 0$ for all $i = 1 \ldots n$, we will be able to prove that pullbacks are weakly preserved, and hence that coalgebraic bisimulation and behavioral equivalence are the same.

Obviously, Milner-type strong bisimulation [21, 22] and bisimulation for *FuTS* over $\mathbb{B}$ coincide. Also, strong bisimulation of [17] involving, apart from the usual transfer conditions, the comparison of state information, viz. the apparent rates, can be treated with *FuTS*. Again the two notions of equivalence coincide. We expect to be able to deal with discrete time and so-called Markov automata as well. For dense time and general measures one may speculate that the use of functions of compact support with respect to a suitable topology may be fruitful. Future research needs to reveal under what algebraic conditions of the semirings, or similar structures, or the coalgebraic conditions on the format of the functors involved standard bisimulation, *FuTS*-bisimulation, coalgebraic bisimulation and behavioral equivalence will amount to similar identifications.

Acknowledgments The authors are grateful to Rocco De Nicola, Michele Loreti and Jan Rutten for fruitful discussions useful suggestions. DL and MM acknowledge support by EU Project n. 257414 Autonomic Service-Components Ensembles (ASCENS) and by CNR/RSTL Project XXL. This research has been conducted while EV was spending a sabbatical leave at the CNR/ISTI. EV gratefully acknowledges the hospitality and support during his stay in Pisa.

References

[1] C. Baier, B.R. Haverkort, H. Hermanns, J.-P. Katoen & M. Siegle, editors (2004): *Validation of Stochastic Systems – A Guide to Current Research*. LNCS 2925, doi:10.1007/b98484.

[2] J.W. de Bakker & E.P. de Vink (1996): *Control Flow Semantics*. The MIT Press.

[3] M. Bernardo (2007): *A Survey of Markovian Behavioral Equivalences*. In M. Bernardo & J. Hillston, editors: *SFM 2007 Advanced Lectures*, LNCS 4486, pp. 180–219, doi:10.1007/978-3-540-72522-0_5.

[4] M. Bernardo & R. Gorrieri (1998): *A tutorial on EMPA: a theory of concurrent processes with nondeterminism, priorities, probabilities and time*. Theoretical Computer Science 202(1–2), pp. 1–54, doi:10.1016/S0304-3975(97)00127-8.

[5] F. Bonchi, M. Bonsangue, M. Boreale, J. Rutten & A. Silva (2011): *A coalgebraic perspective on linear weighted automata*. Technical Report SEN–1104, CWI. 31pp.

[6] M. Boreale (2009): *Weighted Bisimulation in Linear Algebraic Form*. In M. Bravetti & G. Zavattaro, editors: Proc. CONCUR 2009, LNCS 5710, pp. 163–177, doi:10.1007/978-3-642-04081-8_12.

[7] R. De Nicola, D. Latella, M. Loreti & M. Massink (2009): *Rate-based Transition Systems for Stochastic Process Calculi*. In S. Albers et al., editor: Proc. ICALP 2009, Part II, LNCS 5556, pp. 435–446, doi:10.1007/978-3-642-02930-1_36.

[8] R. De Nicola, D. Latella, M. Loreti & M. Massink (2011): *State to function labelled transition systems: a uniform framework for defining stochastic process calculi*. Technical Report ISTI-2011-TR-012, CNR/ISTI.

[9] R. De Nicola, D. Latella & M. Massink (2005): *Formal modeling and quantitative analysis of Klaim-based mobile systems*. In H. Haddad et al., editor: Proc. SAC 2005, ACM, pp. 428–435, doi:10.1145/1066677.1066777.

[10] C. Eisentraut, H. Hermanns & L. Zhang (2010): *Concurrency and Composition in a Stochastic World*. In P. Gastin & F. Laroussinie, editors: Proc. CONCUR 2010, LNCS 6269, pp. 21–39, doi:10.1007/978-3-642-15375-4_3.

[11] H.P. Gumm & T. Schröder (2001): *Products of coalgebras*. Algebra Universalis 46, pp. 163–185.

[12] H.P. Gumm & T. Schröder (2002): *Coalgebras of bounded type*. Mathematical Structures in Computer Science 12, pp. 565–578, doi:10.1017/S0960129501003590.

[13] H. Hermanns (2002): *Interactive Markov Chains*. LNCS 2428, doi:10.1007/3-540-45804-2.

[14] H. Hermanns, U. Herzog & J.-P. Katoen (2002): *Process algebra for performance evaluation*. Theoretical Computer Science 274(1–2), pp. 43–87, doi:10.1016/S0304-3975(00)00305-4.

[15] H. Hermanns, U. Herzog & V. Mertsiotakis (1998): *Stochastic process algebras – between LOTOS and Markov chains*. Computer Networks and ISDN Systems 30, pp. 901–924, doi:10.1016/S0169-7552(97)00133-5.

[16] H. Hermanns & J.-P. Katoen (2010): *The How and Why of Interactive Markov Chains*. In F.S. de Boer, M.M. Bonsangue, S. Hallerstede & M. Leuschel, editors: Proc. FMCO 2009, LNCS 6286, pp. 311–337, doi:10.1007/978-3-642-17071-3_16.

[17] J. Hillston (1996): *A Compositional Approach to Performance Modelling*. Distinguished Dissertations in Computer Science 12, Cambridge University Press.

[18] J. Hillston (2005): *Process Algebras for Quantitative Analysis*. In: Proc. LICS, Chicago, IEEE, pp. 239–248, doi:10.1109/LICS.2005.35.

[19] B. Klin & V. Sassone (2008): *Structural Operational Semantics for Stochastic Process Calculi*. In R.M. Amadio, editor: Proc. FoSSaCS 2008, LNCS 4962, pp. 428–442, doi:10.1007/978-3-540-78499-9_30.

[20] A. Kurz (2000): *Logics for coalgebras and applications to computer science*. Ph.D. thesis, LMU München.

[21] R. Milner (1980): *A Calculus of Communicating Systems*. LNCS 92, doi:10.1007/3-540-10235-3.

[22] D. Park (1981): *Concurrency and Automata on Infinite Sequences*. In: Proc. GI-Conference 1981, Karlsruhe, LNCS 104, pp. 167–183.

[23] C. Priami (1995): *Stochastic π-calculus*. The Computer Journal 38, pp. 578–589, doi:10.1093/comjnl/38.7.578.

[24] J.J.M.M. Rutten (2000): *Universal coalgebra: a theory of systems*. Theoretical Computer Science 249, pp. 3–80, doi:10.1016/S0304-3975(00)00056-6.

[25] A. Silva, F. Bonchi, M. Bonsangue & J. Rutten (2011): *Quantitative Kleene coalgebras*. Information and Computation 209(5), pp. 822–846, doi:10.1016/j.ic.2010.09.007.

[26] A. Sokolova (2011): *Probabilistic systems coalgebraically: a survey*. Theoretical Computer Science 412(38), pp. 5095–5110, doi:10.1016/j.tcs.2011.05.008.

[27] D. Turi & G.D. Plotkin (1997): *Towards a Mathematical Operational Semantics*. In: Proc. LICS 1997, Warsaw, IEEE, pp. 280–291, doi:10.1109/LICS.1999.782615.

[28] E.P. de Vink & J.J.M.M. Rutten (1999): *Bisimulation for probabilistic transition systems: a coalgebraic approach*. Theoretical Computer Science 221, pp. 271–293, doi:10.1016/S0304-3975(99)00035-3.

A Additional proofs

Additional proof for Section 2

Lemma 1. Let X be a set, $\mathcal{R}$ a semiring and $|$ an injective binary operation on X. For $\varphi,\psi \in \mathcal{FS}(X,\mathcal{R})$ it holds that $\oplus(\varphi + \psi) = \oplus\varphi + \oplus\psi$ and $\oplus(\varphi \,|\, \psi) = (\oplus\varphi) * (\oplus\psi)$.

Proof. We verify $\oplus(\varphi \,|\, \psi) = (\oplus\varphi) * (\oplus\psi)$ for arbitrary $\varphi,\psi \in \mathcal{FS}(X,\mathcal{R})$.

$$\begin{array}{lll}
\multicolumn{2}{l}{(\oplus\varphi) * (\oplus\psi)} & \\
= & \big(\sum_{x_1 \in X} \varphi(x_1)\big) * \big(\sum_{x_2 \in X} \psi(x_2)\big) & \text{(by definition } \oplus) \\
= & \sum_{x_1,x_2 \in X} \varphi(x_1) * \psi(x_2) & \text{(by distributivity of } * \text{ over } +) \\
= & \sum_{x \in X,\, x = x_1 | x_2} \varphi(x_1) * \psi(x_2) & \text{(by injectivity of } |\,) \\
= & \sum_{x \in X,\, x = x_1 | x_2} (\varphi \,|\, \psi)(x) & \text{(by definition of } \varphi \,|\, \psi) \\
= & \sum_{x \in X} (\varphi \,|\, \psi)(x) & \text{(since } (\varphi \,|\, \psi)(x) = 0 \text{ if } x \notin |^{-1}(X))
\end{array}$$

The fact $\oplus(\varphi + \psi) = \oplus\varphi + \oplus\psi$ follows direct from the definitions and commutativity of $+$. □

Additional proof for Section 4

Lemma 3. Let $\mathcal{L}$ be a set of labels and $\mathcal{R}$ a semiring. Then functor $\mathcal{V}_{\mathcal{R}}^{\mathcal{L}}$ on **Set** is bounded.

Proof. Consider the elements $\nu \in \mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}$ as parametrized 'valuation' functions, and the elements $\sigma \in X^{\mathbb{N}}$ as 'selection' functions. The functor $\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}} \times (\cdot)^{\mathbb{N}} : \mathbf{Set} \to \mathbf{Set}$ is the product functor of the lifting $id_{\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}}$ of the identity functor $id_{\mathcal{FS}(\mathbb{N},\mathcal{R})}$ to $\mathcal{L}$ and of the functor $(\cdot)^{\mathbb{N}}$. Define the mapping $\eta : id_{\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}} \times (\cdot)^{\mathbb{N}} \to \mathcal{V}_{\mathcal{R}}^{\mathcal{L}}$ by putting

$$\eta_X(\nu,\sigma)(\ell)(x) \;=\; \sum_{n \in \sigma^{-1}(x)} \nu(\ell)(n)$$

for $\nu \in \mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}$, $\sigma \in X^{\mathbb{N}}$ and $\ell \in \mathcal{L}$. For $\ell \in \mathcal{L}$ and $x \in X$, the right-hand sum defining $\eta_X(\nu,\sigma)(\ell)(x)$ exists, since $\nu(\ell) : \mathbb{N} \to \mathcal{R}$ is of finite support. Note that $\eta_X(\nu,\sigma)(\ell)$ is of finite support too: If $\eta_X(\nu,\sigma)(\ell)(x) \neq 0$, by definition $\sum\{\, \nu(\ell)(n) \mid n \in \sigma^{-1}(x) \,\} \neq 0$. Then $\nu(\ell)(n) \neq 0$ for some $n \in \sigma^{-1}(x)$. Thus, $\sigma(n) = x$ for some $n \in spt(\nu(\ell))$. So, $spt(\eta_X(\nu,\sigma)) \subseteq \{\, \sigma(n) \mid n \in spt(\nu(\ell)) \,\}$ and $spt(\eta_X(\nu,\sigma))$ is finite.

Next we verify that $\eta : id_{\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}} \times (\cdot)^{\mathbb{N}} \to \mathcal{V}_{\mathcal{R}}^{\mathcal{L}}$ is a natural transformation, i.e. we check that for $f : X \to Y$ it holds that $\mathcal{V}_{\mathcal{R}}^{\mathcal{L}} \circ \eta_X = \eta_Y \circ \langle\, id_{\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}},\, f^{\mathbb{N}} \,\rangle$.

$$\begin{tikzcd}
\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}} \times X^{\mathbb{N}} \arrow[r, "id \times f^{\mathbb{N}}"] \arrow[d, "\eta_X"'] & \mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}} \times Y^{\mathbb{N}} \arrow[d, "\eta_Y"] \\
\mathcal{FS}(X,\mathcal{R})^{\mathcal{L}} \arrow[r, "\mathcal{FS}(f,\mathcal{R})^{\mathcal{L}}"'] & \mathcal{FS}(Y,\mathcal{R})^{\mathcal{L}}
\end{tikzcd}$$

For $\nu \in \mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}$ and $\sigma \in X^{\mathbb{N}}$ we have, for $\ell \in \mathcal{L}$ and $y \in Y$,

$$\begin{array}{llll}
\multicolumn{4}{l}{(\mathcal{FS}(f,\mathcal{R})^{\mathcal{L}} \circ \eta_X)(\nu,\sigma)(\ell)(y)} \\
= & \sum_{x \in f^{-1}(y)} \eta_X(\nu,\sigma)(\ell)(x) & = & \sum_{x \in f^{-1}(y)} \sum_{n \in \sigma^{-1}(x)} \nu(\ell)(n) \\
= & \sum_{n \in (f \circ \sigma)^{-1}} \nu(\ell)(n) & = & \eta_Y(\nu, f \circ \sigma)(\ell)(y) \\
= & \eta_Y\big((id_{\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}} \times f^{\mathbb{N}})(\nu,\sigma)\big)(\ell)(y) & = & (\eta_Y \circ (id_{\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}} \times f^{\mathbb{N}}))(\nu,\sigma)(\ell)(y)
\end{array}$$

Thus, $\mathcal{FS}(f,\mathcal{R})^{\mathcal{L}} \circ \eta_X = \eta_Y \circ (id_{\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}} \times f^{\mathbb{N}})$ and $\eta : id_{\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}} \times (\cdot)^{\mathbb{N}} \to \mathcal{V}_{\mathcal{R}}^{\mathcal{L}}$ is a natural transformation.

Finally, we check that $\eta_X : id_{\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}} \times X^{\mathbb{N}} \to \mathcal{V}_{\mathcal{R}}^{\mathcal{L}}(X)$ is surjective. Choose a set X and a mapping $\varphi : \mathcal{L} \to \mathcal{FS}(X,\mathcal{R})$. Say, $spt(\varphi(\ell)) = \{ x_1^{\ell}, \ldots, x_{n(\ell)}^{\ell} \}$. Without loss of generality we assume $X \neq spt(\varphi(\ell))$ and pick $x_0^{\ell} \in X \backslash spt(\varphi(\ell))$. Define $\nu \in \mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}$ by $\nu(\ell)(n) = \varphi(\ell)(x_n^{\ell})$ for $n = 1 \ldots n(\ell)$ and $\nu(\ell)(n) = 0$ otherwise. Define $\sigma : \mathbb{N} \to X$ by $\sigma(n) = x_n^{\ell}$ for $1 \leqslant n \leqslant n(\ell)$ and $\sigma(n) = x_0^{\ell}$ otherwise. Then we have

$$\begin{array}{rclclcl}
\eta_X \langle \nu,\sigma \rangle(\ell)(x_i^{\ell}) & = & \sum_{m \in \sigma^{-1}(x_i^{\ell})} \nu(\ell)(m) & = & \nu(\ell)(i) & = & \varphi(\ell)(x_i^{\ell}) \\
\eta_X \langle \nu,\sigma \rangle(\ell)(x) & = & \sum_{m \in \sigma^{-1}(x)} \nu(\ell)(m) & = & \sum_{n \in \mathbb{N} \backslash \{1,\ldots,n(\ell)\}} \nu(\ell)(n) & = & 0
\end{array}$$

for $i = 1 \ldots n(\ell)$ and $x \notin spt(\nu(\ell))$. Thus $\eta_X \langle \nu,\sigma \rangle(\ell)(x) = \varphi(\ell)(x)$ for all $\ell \in \mathcal{L}$ and $x \in X$, $\eta_X \langle \nu,\sigma \rangle = \varphi$ and $\eta_X : id_{\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}} \times X^{\mathbb{N}} \to \mathcal{V}_{\mathcal{R}}^{\mathcal{L}}(X)$ is surjective. □

Additional proofs for Section 5

Definition.[17, Definition 3.3.1] We put

$$\begin{array}{lcll@{\qquad}lcll}
r_a(\mathbf{nil}) & = & 0 & & r_a(P + Q) & = & r_a(P) + r_a(Q) & \\
r_a((a,\lambda).P) & = & \lambda & & r_a(P \parallel_A Q) & = & r_a(P) + r_a(Q) & \text{if } a \notin A \\
r_a((b,\lambda).P) & = & 0 & \text{for } b \neq a & r_a(P \parallel_A Q) & = & \min\{ r_a(P), r_a(Q) \} & \text{if } a \in A \\
r_a(X) & = & r_a(P) & \text{if } X := P & & & &
\end{array}$$

Lemma 6. Let $P \in \mathcal{P}_{PEPA}$ and $a \in \mathcal{A}$. Suppose $P \overset{\delta_a}{\rightarrowtail}_p \mathcal{P}$. Then $\oplus\mathcal{P} = r_a(P)$.

Proof. Guarded recursion. We treat the two cases for the parallel construct.

Case $P = P_1 \parallel_A P_2$, $a \notin A$. Suppose $P_1 \overset{\delta_a}{\rightarrowtail}_p \mathcal{P}_1$, $P_2 \overset{\delta_a}{\rightarrowtail}_p \mathcal{P}_2$. Then $\mathcal{P} = (\mathcal{P}_1 \parallel_A \mathcal{X}_{P_2}) + (\mathcal{X}_{P_1} \parallel_A \mathcal{P}_2)$. Therefore we have

$$\begin{array}{rcll}
\oplus\mathcal{P} & = & \oplus(\mathcal{P}_1 \parallel_A \mathcal{X}_{P_2}) + (\mathcal{X}_{P_1} \parallel_A \mathcal{P}_2) & \\
 & = & \oplus(\mathcal{P}_1 \parallel_A \mathcal{X}_{P_2}) + \oplus(\mathcal{X}_{P_1} \parallel_A \mathcal{P}_2) & \text{(by Lemma 1)} \\
 & = & (\oplus\mathcal{P}_1 \cdot \oplus\mathcal{X}_{P_2}) + (\oplus\mathcal{X}_{P_1} \cdot \oplus\mathcal{P}_2) & \text{(by Lemma 1)} \\
 & = & \oplus\mathcal{P}_1 + \oplus\mathcal{P}_2 & \text{(since } \oplus\mathcal{X}_{P_1}, \oplus\mathcal{X}_{P_2} = 1) \\
 & = & r_a(P_1) + r_a(P_2) & \text{(by the induction hypothesis)} \\
 & = & r_a(P_1 \parallel_A P_2) & \text{(by definition } r_a)
\end{array}$$

Case $P = P_1 \parallel_A P_2$, $a \in A$. Suppose $P_1 \overset{\delta_a}{\rightarrowtail}_p \mathcal{P}_1$, $P_2 \overset{\delta_a}{\rightarrowtail}_p \mathcal{P}_2$. Then $\mathcal{P} = arf(\mathcal{P}_1,\mathcal{P}_2) \cdot (\mathcal{P}_1 \parallel_A \mathcal{P}_2)$. If $\oplus\mathcal{P}_1, \oplus\mathcal{P}_2 > 0$ we have

$$\begin{array}{rcll}
\oplus\mathcal{P} & = & \oplus(arf(\mathcal{P}_1,\mathcal{P}_2) \cdot (\mathcal{P}_1 \parallel_A \mathcal{P}_2) & \\
 & = & arf(\mathcal{P}_1,\mathcal{P}_2) \cdot \oplus\mathcal{P}_1 \cdot \oplus\mathcal{P}_2 & \text{(by Lemma 1)} \\
 & = & \dfrac{\min\{ \oplus\mathcal{P}_1, \oplus\mathcal{P}_2 \}}{\oplus\mathcal{P}_1 \cdot \oplus\mathcal{P}_2)} \cdot \oplus\mathcal{P}_1 \cdot \oplus\mathcal{P}_2 & \text{(by definition of } arf) \\
 & = & \min\{ \oplus\mathcal{P}_1, \oplus\mathcal{P}_2 \} & \\
 & = & \min\{ r_a(P_1), r_a(P_2) \} & \text{(by the induction hypothesis)} \\
 & = & r_a(P_1 \parallel_A P_2) & \text{(by definition } r_a)
\end{array}$$

If $\oplus\mathcal{P}_1,\oplus\mathcal{P}_2 = 0$, then $\mathit{arf}(\mathcal{P}_1,\mathcal{P}_2) = 0$, by definition, and $r_a(P_1), r_a(P_2) = 0$, by induction hypothesis. Therefore we have $\oplus\mathcal{P} = \mathit{arf}(\mathcal{P}_1,\mathcal{P}_2)\cdot\oplus\mathcal{P}_1\cdot\oplus\mathcal{P}_2 = 0$ as well as $r_a(P_1 \parallel_A P_2) = \min\{r_a(P_1), r_a(P_2)\} = 0$. So, also now, $\oplus\mathcal{P} = r_a(P_1 \parallel_A P_2)$. The other cases are straightforward, in the case of $P_1 + P_2$ also relying on Lemma 1. □

Corollary. If $P \overset{\delta_a}{\rightarrowtail} \mathcal{P}$ and $Q \overset{\delta_a}{\rightarrowtail} \mathcal{Q}$, then $\mathit{arf}(\mathcal{P},\mathcal{Q}) = \mathit{arf}(P,Q)$.

Proof. Direct from the definitions. □

Lemma 8. Let $P \in \mathcal{P}_{PEPA}$ and $a \in \mathcal{A}$. Suppose $P \overset{\delta_a}{\rightarrowtail} \mathcal{P}$. Then it holds that $\mathcal{P}(P') = \sum \{\!|\, \lambda \mid P \overset{a,\lambda}{\rightarrow} P' \,|\!\}$ for all $P' \in \mathcal{P}_{PEPA}$.

Proof. Guarded induction on P. We only treat the cases for the parallel composition. Note, the operation $\parallel_A : \mathcal{P}_{PEPA} \times \mathcal{P}_{PEPA} \to \mathcal{P}_{PEPA}$ with $\parallel_A(P_1,P_2) = P_1 \parallel_A P_2$ is injective. Recall, for $\mathcal{P}_1,\mathcal{P}_2 \in \mathcal{FS}(\mathcal{P}_{PEPA},\mathbb{R}_{\geqslant 0})$, we have $(\mathcal{P}_1 \parallel_A \mathcal{P}_2)(P_1 \parallel_A P_2) = \mathcal{P}_1(P_1)\cdot\mathcal{P}_2(P_2)$.

Suppose $a \notin \mathcal{A}$. Assume $P_1 \overset{a}{\rightarrowtail} \mathcal{P}_1$, $P_2 \overset{a}{\rightarrowtail} \mathcal{P}_2$, $P_1 \parallel_A P_2 \overset{a}{\rightarrowtail} \mathcal{P}$. We distinguish three cases. Case (I), $P' = P'_1 \parallel_A P_2$, $P'_1 \neq P_1$. Then we have

$$\begin{aligned}
&\textstyle\sum \{\!|\, \lambda \mid P_1 \parallel_A P_2 \overset{a,\lambda}{\rightarrow} P' \,|\!\} \\
&= \textstyle\sum \{\!|\, \lambda \mid P_1 \overset{a,\lambda}{\rightarrow} P'_1 \,|\!\} && \text{(by rule (PAR1a))} \\
&= \mathcal{P}_1(P'_1) && \text{(by the induction hypothesis)} \\
&= \mathcal{P}_1(P'_1)\cdot\mathcal{X}_{P_2}(P_2) && \text{(as } \mathcal{X}_{P_2}(P_2) = 1\text{)} \\
&= (\mathcal{P}_1 \parallel_A \mathcal{X}_{P_2})(P'_1 \parallel_A P_2) + (\mathcal{X}_{P_1} \parallel_A \mathcal{P}_2)(P'_1 \parallel_A P_2) \\
& && \text{(definition } \parallel_A \text{ on } \mathcal{FS}(\mathcal{P}_{PEPA},\mathbb{R}_{\geqslant 0}),\ \mathcal{X}_{P_1}(P'_1) = 0\text{)} \\
&= \mathcal{P}(P') && \text{(by rule (PAR1)}
\end{aligned}$$

Case (II), $P' = P_1 \parallel_A P'_2$, $P'_2 \neq P_2$: similar. Case (III), $P' = P_1 \parallel_A P_2$. Then we have

$$\begin{aligned}
&\textstyle\sum \{\!|\, \lambda \mid P_1 \parallel_A P_2 \overset{a,\lambda}{\rightarrow} P' \,|\!\} \\
&= \left(\textstyle\sum \{\!|\, \lambda \mid P_1 \overset{a,\lambda}{\rightarrow} P_1 \,|\!\}\right) + \left(\textstyle\sum \{\!|\, \lambda \mid P_2 \overset{a,\lambda}{\rightarrow} P_2 \,|\!\}\right) && \text{(by rules (PAR1a) and (PAR1b))} \\
&= \mathcal{P}_1(P_1) + \mathcal{P}_2(P_2) && \text{(by the induction hypothesis)} \\
&= (\mathcal{P}_1 \parallel_A \mathcal{X}_{P_2})(P_1 \parallel_A P_2) + (\mathcal{X}_{P_1} \parallel_A \mathcal{P}_2)(P_1 \parallel_A P_2) \\
& && \text{(definition } \parallel_A \text{ on } \mathcal{FS}(\mathcal{P}_{PEPA},\mathbb{R}_{\geqslant 0}),\ \mathcal{X}_{P_1}(P_1),\ \mathcal{X}_{P_2}(P_2) = 1\text{)} \\
&= \mathcal{P}(P') && \text{(again by rule (PAR1)}
\end{aligned}$$

Suppose $a \in A$. Assume $P_1 \overset{a}{\rightarrowtail} \mathcal{P}_1$, $P_2 \overset{a}{\rightarrowtail} \mathcal{P}_2$, $P_1 \parallel_A P_2 \overset{a}{\rightarrowtail} \mathcal{P}$. Without loss of generality, $P' = P'_1 \parallel_A P'_2$ for suitable $P'_1, P'_2 \in \mathcal{P}_{PEPA}$.

$$\begin{array}{lll}
\multicolumn{3}{l}{\sum \{\!| \lambda \mid P_1 \parallel_A P_2 \xrightarrow{a,\lambda} P' |\!\}} \\
= & \sum \{\!| \mathit{arf}(P_1,P_2) \cdot \lambda_1 \cdot \lambda_2 \mid P_1 \xrightarrow{a,\lambda_1} P_1', P_2 \xrightarrow{a,\lambda_2} P_2' |\!\} & \text{(by rule (PAR2))} \\
= & \mathit{arf}(P_1,P_2) \cdot \left(\sum \{\!| \lambda_1 \mid P_1 \xrightarrow{a,\lambda_1} P_1' |\!\} \right) \cdot \left(\sum \{\!| \lambda_2 \mid P_2 \xrightarrow{a,\lambda_2} P_2' |\!\} \right) & \text{(by distributivity)} \\
= & \mathit{arf}(P_1,P_2) \cdot \mathcal{P}_1(P_1') \cdot \mathcal{P}_2(P_2') & \text{(by the induction hypothesis)} \\
= & \mathit{arf}(\mathcal{P}_1,\mathcal{P}_2) \cdot \mathcal{P}_1(P_1') \cdot \mathcal{P}_2(P_2') & \text{(by the corollary above)} \\
= & \mathit{arf}(\mathcal{P}_1,\mathcal{P}_2) \cdot (\mathcal{P}_1 \parallel_A \mathcal{P}_2)(P_1' \parallel_A P_2') & \text{(definition } \parallel_A \text{ on } \mathcal{FS}(\mathcal{P}_{PEPA}, \mathbb{R}_{\geqslant 0})) \\
= & \mathcal{P}(P') & \text{(by rule (PAR2)}
\end{array}$$

The other cases are simpler and omitted here. □

Additional proofs for Section 6

Lemma 11.

(a) Let $P \in \mathcal{P}_{IML}$ and $a \in \mathcal{A}$. If $P \overset{a}{\rightarrowtail}_1 \mathcal{P}$ then $P \xrightarrow{a} P' \iff \mathcal{P}(P') = \mathtt{true}$.

(b) Let $P \in \mathcal{P}_{IML}$. If $P \overset{\delta}{\rightarrowtail}_2 \mathcal{P}$ then $\sum \{\!| \lambda \mid P \overset{\lambda}{\dashrightarrow} P' |\!\} = \mathcal{P}(P')$. □

Proof. (a) Guarded induction. Let $a \in \mathcal{A}$. We treat the typical cases $\lambda.P$ and $P_1 \parallel_A P_2$ for $a \notin A$.

Case $\lambda.P$. Suppose $\lambda.P \overset{a}{\rightarrowtail} \mathcal{P}$. Then we have $\mathcal{P} = []_{\mathbb{B}}$. Thus, both $\lambda.P \xrightarrow{a} P'$ for no $P' \in \mathcal{P}_{IML}$, as no transition is provided in $\rightarrow$, and $\mathcal{P}(P') = \mathtt{false}$ by definition of $[]_{\mathbb{B}}$, for all $P' \in \mathcal{P}_{IML}$.

Case $P_1 \parallel_A P_2$, $a \notin A$. Suppose $P_1 \overset{a}{\rightarrowtail} \mathcal{P}_1$, $P_2 \overset{a}{\rightarrowtail} \mathcal{P}_2$ and $P_1 \parallel_A P_2 \overset{a}{\rightarrowtail} \mathcal{P}$. Then it holds that $\mathcal{P} = (\mathcal{P}_1 \parallel_A \mathcal{X}_{P_2}) + (\mathcal{X}_{P_1} \parallel_A \mathcal{P}_2)$. Recall, for $Q \in \mathcal{P}_{IML}$ and $\mathcal{X}_Q \in \mathcal{FS}(\mathcal{P}_{IML}, \mathbb{B})$, $\mathcal{X}_Q(Q') = \mathtt{true}$ iff $Q' = Q$, for $Q' \in \mathcal{P}_{IML}$. We have

$$\begin{array}{ll}
\multicolumn{2}{l}{P_1 \parallel_A P_2 \xrightarrow{a} P'} \\
\Leftrightarrow & (P_1 \xrightarrow{a} P_1' \wedge P' = P_1' \parallel_A P_2) \vee (P_2 \xrightarrow{a} P_2' \wedge P' = P_1 \parallel_A P_2') \\
 & \text{(by analysis of } \rightarrow\text{)} \\
\Leftrightarrow & (\mathcal{P}_1(P_1') = \mathtt{true} \wedge P' = P_1' \parallel_A P_2) \vee (\mathcal{P}_2(P_2') = \mathtt{true} \wedge P' = P_1 \parallel_A P_2') \\
 & \text{(by the induction hypothesis)} \\
\Leftrightarrow & (\mathcal{P}_1(P_1') \cdot \mathcal{X}_{P_2}(P_2) = \mathtt{true} \wedge P' = P_1' \parallel_A P_2) \vee (\mathcal{X}_{P_1}(P_1) \cdot \mathcal{P}_2(P_2') = \mathtt{true} \wedge P' = P_1 \parallel_A P_2') \\
 & \text{(by definition of } \mathcal{X}_{P_1} \text{ and } \mathcal{X}_{P_2}\text{)} \\
\Leftrightarrow & ((\mathcal{P}_1 \parallel_A \mathcal{X}_{P_2})(P_1' \parallel_A P_2) = \mathtt{true} \wedge P' = P_1' \parallel_A P_2) \\
 & \qquad \vee ((\mathcal{X}_{P_1} \parallel_A \mathcal{P}_2)(P_1 \parallel_A P_2') = \mathtt{true} \wedge P' = P_1 \parallel_A P_2') \\
 & \text{(by definition of } \parallel_A\text{)} \\
\Leftrightarrow & (\mathcal{P}_1 \parallel_A \mathcal{X}_{P_2})(P') = \mathtt{true} \vee (\mathcal{X}_{P_1} \parallel_A \mathcal{P}_2)(P') = \mathtt{true} \\
 & \text{(by definition of } \parallel_A, \mathcal{X}_{P_1} \text{ and } \mathcal{X}_{P_2}\text{)} \\
\Leftrightarrow & ((\mathcal{P}_1 \parallel_A \mathcal{X}_{P_2}) + (\mathcal{X}_{P_1} \parallel_A \mathcal{P}_2))(P') = \mathtt{true} \\
 & \text{(by definition of } + \text{ on } \mathcal{FS}(\mathcal{P}_{IML}, \mathbb{B})\text{)} \\
\Leftrightarrow & \mathcal{P}(P') = \mathtt{true}
\end{array}$$

The other cases are standard or similar and easier.

(b) Guarded induction. We treat the cases for $\mu.P$ and $P_1 \parallel_A P_2$. Case $\mu.P$. Assume $P \overset{\delta}{\rightarrowtail}_2 \mathcal{P}$. Suppose $P = \mu.P'$. Then it holds that P admits a single $\dashrightarrow$-transition, viz. $P \overset{\mu}{\dashrightarrow} P'$. Thus we have $\sum \{\!|\, \lambda \mid P \overset{\lambda}{\dashrightarrow} P' \,|\!\} = \mu = [\, P' \mapsto \mu\,](P') = \mathcal{P}(P')$. Suppose $P = \mu.P''$ for some $P'' \neq P$. Then we have $\sum \{\!|\, \lambda \mid P \overset{\lambda}{\dashrightarrow} P' \,|\!\} = 0 = [\, P'' \mapsto \mu\,](P') = \mathcal{P}(P')$.

Case $P_1 \parallel_A P_2$. Assume $P_1 \overset{\delta}{\rightarrowtail}_2 \mathcal{P}_1$, $P_2 \overset{\delta}{\rightarrowtail}_2 \mathcal{P}_2$ and $P_1 \parallel_A P_2 \overset{\delta}{\rightarrowtail} \mathcal{P}$. It holds that $\mathcal{P} = (\mathcal{P}_1 \parallel_A \mathcal{X}_{P_2}) + (\mathcal{X}_{P_1} \parallel_A \mathcal{P}_2)$. We calculate

$$
\begin{aligned}
&\textstyle\sum \{\!|\, \lambda \mid P_1 \parallel_A P_2 \overset{\lambda}{\dashrightarrow} P' \,|\!\} \\
=\;& \textstyle\sum \{\!|\, \lambda \mid P_1 \overset{\lambda}{\dashrightarrow} P_1',\ P' = P_1' \parallel_A P_2 \,|\!\} + \sum \{\!|\, \lambda \mid P_2 \overset{\lambda}{\dashrightarrow} P_2',\ P' = P_1 \parallel_A P_2' \,|\!\} \\
&\text{(by analysis of } \dashrightarrow\text{)} \\
=\;& (\texttt{if}\ P' = P_1' \parallel_A P_2\ \texttt{then} \textstyle\sum \{\!|\, \lambda \mid P_1 \overset{\lambda}{\dashrightarrow} P_1' \,|\!\}\ \texttt{else}\ 0\ \texttt{end}) + \\
&\qquad\qquad (\texttt{if}\ P' = P_1 \parallel_A P_2'\ \texttt{then} \textstyle\sum \{\!|\, \lambda \mid P_2 \overset{\lambda}{\dashrightarrow} P_2' \,|\!\}\ \texttt{else}\ 0\ \texttt{end}) \\
=\;& (\texttt{if}\ P' = P_1' \parallel_A P_2\ \texttt{then}\ \mathcal{P}_1(P_1')\ \texttt{else}\ 0\ \texttt{end}) + \\
&\qquad\qquad (\texttt{if}\ P' = P_1 \parallel_A P_2'\ \texttt{then}\ \mathcal{P}_2(P_2')\ \texttt{else}\ 0\ \texttt{end}) \\
&\text{(by induction hypothesis for } P_1 \text{ and } P_2\text{)} \\
=\;& (\mathcal{P}_1 \parallel_A \mathcal{X}_{P_2})(P') + (\mathcal{X}_{P_1} \parallel_A \mathcal{P}_2)(P') \\
&\text{(by definition of } \parallel_A,\ \mathcal{X}_{P_1}, \mathcal{X}_{P_2} \text{ and } + \text{ on } \mathcal{FS}(\mathcal{P}_{IML}, \mathbb{R}_{\geqslant 0})\text{)} \\
=\;& \mathcal{P}(P')
\end{aligned}
$$

The remaining cases are left to the reader. □