www.ascens-ist.eu

ASCENS

Autonomic Service-Component Ensembles

TR 09: Coalgebraic Bisimulation of FuTS

Grant agreement number: **257414**
Funding Scheme: **FET Proactive**
Project Type: **Integrated Project**
Latest version of Annex I: **7.6.2010**

Author(s): **D. Latella (ISTI), M. Massink (ISTI), E. P. de Vink (Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven and Centrum Wiskunde en Informatica, Amsterdam)**

Date of technical report: **January 16, 2013**
Revision: **V1**
Classification: **PU**

Project coordinator: **Martin Wirsing (LMU)**
Tel: **+49 89 2180 9154**
Fax: **+49 89 2180 9175**
E-mail: **wirsing@lmu.de**

Partners: **LMU, UNIPI, UDF, Fraunhofer, UJF-Verimag, UNIMORE, ULB, EPFL, VW, Zimory, UL, IMT, Mobsya, CUNI**

SEVENTH FRAMEWORK PROGRAMME

Coalgebraic Bisimulation of FuTS

January 16, 2013

abstract

Labeled state-to-function transition systems, *FuTS*s for short, capture transition schemes incorporating multiplicities from states to functions of finite support over general semirings. As such *FuTS*s constitute a convenient modeling instrument to deal with process languages and their stochastic extensions in particular. In this paper, the notion of bisimulation induced by a *FuTS* is addressed from a coalgebraic point of view. A correspondence result is established stating that *FuTS*-bisimilarity coincides with behavioural equivalence of the associated functor. Moreover, it is shown that for *FuTS*s involving a specific type of semiring only, weak pullbacks are preserved. As a consequence, for these *FuTS*s, behavioural equivalence coincides with coalgebraic bisimilarity. As generic examples, the equivalences underlying the stochastic process algebras *PEPA* and *IML* are related to the bisimilarity of specific *FuTS*s. By the correspondence result coalgebraic justification of the equivalences of these calculi is obtained. Further illustrations of *FuTS* semantics are discussed for deterministically (discrete) timed process algebras and Markov Automata.

Contents

1 Introduction

In the last couple of decades, qualitative process description languages have been enriched with quantitative information. In the qualitative case, process description languages equipped with formal operational semantics have proven to be successful formalisms for the modelling of concurrent systems and the analysis of their behaviour. Generally, the operational semantics of a qualitative process description language is given by means of a labelled transition system (*LTS*), with states being process terms and actions and interactions decorating the transitions between states. Typically, based on the induced transition system relation, a notion of process equivalence is defined, providing means to compare systems and to reduce their representation to enhance subsequent verification.

Extensions of qualitative description languages allowing a stochastic representation of time, usually referred to as stochastic process algebras, or stochastic process calculi (*SPC*s), are one of the quantitative enrichments of process languages that have received particular attention. For *SPC*s the main aim has been the integration of qualitative descriptions with quantitative ones in a single mathematical framework, building on the combination of *LTS*s and continuous-time Markov chains. The latter is one of the most successful approaches to modelling and performance analysis of (computer) systems and networks. An overview of *SPC*s, equivalences and related analysis techniques can be found in [Hermanns et al.(2002), Baier et al.(2004), Bernardo(2007)], for example. A common feature of many *SPC*s is that actions are augmented with the rates of exponentially distributed random variables that characterise their duration. Alternatively, actions are assumed to be instantaneous, in which case random variables are used for modelling *delays*, as in [Hermanns(2002)]. Although exploiting the same class of distributions, the models and techniques underlying the definition of the calculi turn out to be significantly different in many respects. A prominent difference concerns the modelling 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 different approaches have been proposed for handling transition multiplicity. The proposals range from multi-relations [Hillston(1996), Hermanns(2002)], to proved transition systems [Priami(1995)], to *LTS*s with numbered transitions [van Glabbeek et al.(1995), Hermanns et al.(2002)], to unique rate names [De Nicola et al.(2005)], just to mention a few.

In [De Nicola et al.(2009)], Latella, Massink et al. proposed a variant of *LTS*s, called Rate Transition Systems (*RTS*s). In *LTS*s, 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 a transition labelled with α. In *RTS*s, a transition is a triple of the form $(P, \alpha, \mathscr{P})$. The first and second component are the source state and the label of the transition, as in *LTS*s, while the third component $\mathscr{P}$ 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 characterising the time for the execution of the action represented by α, necessary to reach P' from P via the transition. If $\mathscr{P}$ maps P' to 0, then state P' cannot be reached from P via this transition with label α. The use of continuation functions provides a clean and simple solution to the transition multiplicity problem and make *RTS*s particularly suited for *SPC* semantics. In order to provide a uniform account of the many *SPC*s proposed in the literature, in previous joint work of the first two authors [De Nicola et al.(2011)] State-to-Function Labelled Transition Systems (*FuTS*s) have been introduced as a natural generalisation of *RTS*s. In *FuTS*s the codomain 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. Furthermore, *FuTS*s are equipped with a rich set of (generic) operations on continuation functions, which makes the framework very well suited for the *compositional* definition of the operational semantics of process calculi, including *SPC*s and models where both non-deterministic behaviour and stochastic delays are modelled, like in the Language of Interactive Markov Chains [Hermanns(2002)]. Finally, *FuTS*s are equipped with a natural notion of bisimilarity which, as we will see, coincides for the concrete cases we studied with the notion of process (strong) equivalences reported in the literature.

In this paper we present a coalgebraic treatment of *FuTS*s that allows 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 total functions from S to $\mathcal{R}_i$ of finite support, a sub-collection of functions also occurring in other work combining coalgebra and quantitative modelling. We will see that that $\mathcal{S}$ is a coalgebra of the product of the functors $\mathcal{FS}(\cdot, \mathcal{R}_i)^{\mathcal{L}_i}$. For this to work, we need the relations $\rightarrowtail_i$ to be total and deterministic for the coalgebraic modelling as a function. Maybe surprisingly, this is not a severe restriction at all in the presence of continuation functions: as we will see, the zero-continuation function, which maps every s' to 0 will be associated to a state s and a transition, in order to indicate that no state s' is reachable from s via that transition, in the usual *LTS*-sense; if s allows a transition to some state s_1 as well as to a state s_2, then the continuation function will simply yield a non-zero value for s_1 and for s_2. Therefore, it is no essential limitation to restrict our investigations to total and deterministic *FuTS*s. For example, by using boolean functions, we can model non-deterministic behaviour, as done in Section 5 and Section 7.

The notion of $\mathcal{S}$-bisimilarity that arises from a *FuTS* $\mathcal{S}$ is reinterpreted coalgebraically in the present paper. Following a familiar argument, we first prove that the functor associated with a *FuTS* possesses a final coalgebra and therefore has an associated notion of behavioural equivalence. Then it is shown that behavioural equivalence of the functor induced by $\mathcal{S}$ coincides with bisimilarity for *FuTS*. Pivotal for the proof is the absence of multiplicities in the *FuTS* treatment of quantities at the level of the transitions. In fact, quantities are accumulated in the function values of the continuations and hidden at the higher level of abstraction. It is noted, in the presence of a final coalgebra for *FuTS* a more general definition of behavioural equivalence based on cospans coincides with the one given here, cf. [Kurz(2000)]. The relationship with coalgebraic bisimulation is also investigated and we prove that, under the condition that the underlying semirings admit a (right) multiplicative inverse for non-zero elements, and satisfy the zero-sum property, i.e. a sum in the semiring is equal to zero if and only if all summands are zero, the functors associated to *FuTS*s preserve weak pullbacks. Consequently, by exploiting a general result on coalgebras, we get that for these functors behavioural equivalence coincides with coalgebraic bisimilarity.

Using the bridge established by the correspondence results, we continue by showing for two well-known stochastic process algebras, viz. Hillston's *PEPA* [Hillston(1996)] and Hermanns's *IML* [Hermanns(2002)], that their standard notion of strong equivalence and strong bisimilarity coincide with bisimilarity of the associated *FuTS* (and thus with behavioural equivalence and coalgebraic bisimilarity of the corresponding functor). *PEPA* stands out as one of the prominent Markovian process algebras, and *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*s 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*s to the multi-relation-based explicit treatment of *PEPA* and *IML*. It is noted that in our treatment below we restrict to the key-fragment of these two *SPC*s.

We finally discuss how *FuTS*s can be used also for the definition of the semantics of deterministically discrete timed process algebras as well as for models which incorporate at the same time non-determinism, discrete probabilities and Markovian randomised delays, like it is the case in Markov

Automata as presented in [Eisentraut et al.(2010a), Eisentraut et al.(2010b)].

Related work on coalgebra includes the papers [de Vink & Rutten(1999), Klin & Sassone(2008)] and [Sokolova(2011)]. These papers 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 [Turi & Plotkin(1997)] 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 [Boreale(2009), Klin(2009), Silva et al.(2011), Bonchi et al.(2011)]. Klin considers weighted transition systems, labeled transition systems that assign a weight to each transition. For commutative monoids the notion of a weighted transition system compares with our notion of a *FuTS*, and for which, when casted to the coalgebraic setting, the associated concept of bisimulation coincides with observational equivalence. Building on a result on bounded functors [Gumm & Schröder(2001)], it is shown in [Bonchi et al.(2011)] for a functor involving functions of finite support over a field that the final coalgebra exists. In the present paper, we have followed the scheme of [Bonchi et al.(2011)] to obtain such a result for a functor induced by a *FuTS*. The process languages with stochastic delays we consider in the sequel, based on *PEPA* and *IML*, involve a multi-way CSP-like parallel operator; components proceed simultaneously when synchronization on an action from the synchronization alphabet that indexes the parallel operator is possible. However, here we do not distinguish between internal and external non-determinism, cf. [Hoare(1985)], as an explicit representation of such a distinction is neither contemplated in *PEPA*, nor in *IML*. A coalgebraic treatment of this distinction is proposed in [Wolter(2002)], which uses a functor for so-called non-deterministic filter automata, viz. $\mathcal{P}(\mathcal{P}(\mathcal{A})) \times [\, \mathcal{A} \nrightarrow \mathcal{P}_f(\cdot)\,]$ involving partial functions from a set of actions $\mathcal{A}$ to a finite power-set. Via currying, this can be brought in the form $\mathcal{FS}(\cdot, \mathbb{B})^{\mathcal{L}}$ for $\mathcal{L} = \mathcal{P}(\mathcal{P}(\mathcal{A})) \times \mathcal{A}$, fitting in the format of the functor for the *FuTS*s considered here. In [Boreale & Gadducci(2006)] processes are interpreted as formal power-series over a semiring in the style of [Rutten(2003)]. This allows to compare testing equivalence for a CSP-style language and bisimulation in a Moore automaton. It is noted, that the notions of equivalence addressed in this paper, as often in coalgebraic treatments of process relations, are all strong bisimilarities.

Structure of the paper The present paper is organised as follows: Section 2 briefly discusses some material on semirings and coalgebras. *FuTS*s as well as the associated notion of bisimulation are presented in Section 3. The coalgebraic counterparts of *FuTS*s and *FuTS*-bisimilarity are defined in Section 4, where we also establish the correspondence with behavioural equivalence of the final coalgebra and with coalgebraic bisimilarity. As a stepping stone towards the treatment of *PEPA* and *IML*, we discuss in Section 5 an elementary process language that constitutes the qualitative core of the two *SPC*s. In Section 6 the standard equivalence of *PEPA* is identified with the bisimulation of a *FuTS* and, hence, with behavioural equivalence and coalgebraic bisimilarity. In Section 7 the same is done for the language of *IMC*s where actions and delays are present on equal footing. A discussion of our results and possible extensions is presented in Section 8. Finally, Section 9 wraps up and discusses directions of future research.

An extended abstract of part of this manuscript has appeared as [Latella et al.(2012)]. The additional contributions of the present paper include a detailed proof of the existence of the final coalgebra of the relevant functors and the investigation on the relationship between *FuTS* bisimilarity and behavioural equivalence, on one side, and coalgebraic bisimilarity, on the other. In particular, it is shown that the functor type involved preserves weak pullbacks when the underlying semiring, amongst other, satisfy the zero-sum property. As illustration we insert the modelling of a basic qualitative process language with *FuTS*s. Finally, we provide a discussion of the *FuTS*-based approach and its coalgebraic view to deterministically discrete timed process algebras and Markov Automata.

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 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. Finally, for $\varphi : X \to \mathcal{R}$, and $X' \subseteq X$, we let $\varphi[X'] = \{ \varphi(x) \mid x \in X' \}$.

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, for $i = 1 \ldots n$, $x_i \in X$ all distinct and $r_i \in \mathcal{R}$, 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 $\mathbf{D}_{\mathcal{R},x} = [\, x \mapsto 1 \,]$ is the Dirac function on $\mathcal{R}$ for $x \in X$; in the sequel we will often drop the subscript $\mathcal{R}$ from $[]_{\mathcal{R}}$ and $\mathbf{D}_{\mathcal{R},x}$, when the semiring is clear from the context.

We use $\oplus\varphi$ 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 $\varphi \mid \psi : X \to \mathcal{R}$, by $(\varphi \mid \psi)(x) = \varphi(x_1) * \psi(x_2)$ if $x = x_1 \mid x_2$ for some $x_1, x_2 \in X$, and $(\varphi \mid \psi)(x) = 0$ otherwise. Injectivity of the operation $|$ guarantees that $\varphi \mid \psi$ is well-defined. Again, $\varphi \mid \psi$ is of finite support as φ and ψ are. This is used in the setting of syntactic processes P that may have the form $P_1 \mid P_2$ for two processes P_1 and P_2 and a syntactic operator $|$. We have the following properties.

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 $\oplus(\varphi \mid \psi) = (\oplus\varphi) * (\oplus\psi)$.* □

We recall some basic definitions from coalgebra. See e.g. [Rutten(2000)] for more details. For a functor $\mathcal{F} : \mathbf{Set} \to \mathbf{Set}$ on the category $\mathbf{Set}$ 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 $\mathcal{F}(f) \circ \alpha = \beta \circ f$. See Figure 1. An $\mathcal{F}$-coalgebra $(\Omega_{\mathcal{F}}, \omega_{\mathcal{F}})$ is called final, if there exists, for every $\mathcal{F}$-coalgebra $\mathcal{S} = (X, \alpha)$, a unique homomorphism $[\![\cdot]\!]^{\mathcal{S}}_{\mathcal{F}} : (X, \alpha) \to (\Omega_{\mathcal{F}}, \omega_{\mathcal{F}})$. Two elements x_1, x_2 of a $\mathcal{F}$-coalgebra $\mathcal{S} = (X, \alpha)$ are called behavioural equivalent with respect to $\mathcal{F}$ if $[\![x_1]\!]^{\mathcal{S}}_{\mathcal{F}} = [\![x_2]\!]^{\mathcal{S}}_{\mathcal{F}}$, denoted $x_1 \approx^{\mathcal{S}}_{\mathcal{F}} x_2$. In the notation $[\![\cdot]\!]^{\mathcal{S}}_{\mathcal{F}}$ as well as $\approx^{\mathcal{S}}_{\mathcal{F}}$, the indication of the specific coalgebra $\mathcal{S}$ will be omitted, when clear from the context.

Using a characterisation of [Gumm & Schröder(2002)], a functor $\mathcal{F}$ on $\mathbf{Set}$ 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 $f^B(\varphi)(b) = f(\varphi(b))$.

In order to deal with product functors in the sequel, we will use the following lemma, where $B_1 + \cdots + B_n$ is the disjoint union of $B_1, \ldots, B_n$ and $\upharpoonright$ denotes restriction on functions.

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "\alpha"'] & Y \arrow[d, "\beta"] \\ \mathcal{F}(X) \arrow[r, "\mathcal{F}(f)"'] & \mathcal{F}(Y) \end{tikzcd} \qquad \begin{tikzcd} X \arrow[d, "\alpha"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\gamma"] & Y \arrow[d, "\beta"] \\ \mathcal{F}(X) & \mathcal{F}(R) \arrow[l, "\mathcal{F}(\pi_1)"] \arrow[r, "\mathcal{F}(\pi_2)"'] & \mathcal{F}(Y) \end{tikzcd}$$

Figure 1: Diagrams of coalgebra morphism and coalgebraic bisimulation

Lemma 2 *Let $A_1, \ldots, A_n$, $B_1, \ldots, B_n$ be sets and $\mathcal{F}_1, \ldots \mathcal{F}_n$ be functors on* **Set**. *Suppose $\eta^i : A_i \times (\cdot)^{B_i} \Rightarrow \mathcal{F}_i$ is a natural transformation such that $\eta^i_X : A_i \times X^{B_i} \to \mathcal{F}_i(X)$, $i = 1 \ldots n$, is surjective. Then $\eta : A_1 \times \cdots \times A_n \times (\cdot)^{B_1 + \cdots + B_n} \Rightarrow \mathcal{F}_1 \times \cdots \times \mathcal{F}_n$ with $\eta_X : A_1 \times \cdots \times A_n \times (X)^{B_1 + \cdots + B_n} \to \mathcal{F}_1(X) \times \cdots \times \mathcal{F}_n(X)$ such that*

$$\eta_X(\langle a_1, \ldots, a_n \rangle, \varphi) = \langle \eta^1_X(a_1, \varphi \restriction B_1), \ \ldots, \ \eta^n_X(a_n, \varphi \restriction B_n) \rangle$$

is a natural transformation. Moreover, for each set X, the mapping η_X is surjective. □

The lemma can be straightforwardly checked. Thus, a product functor $\mathcal{F}_1 \times \cdots \times \mathcal{F}_n$ on **Set**, for which each factor $\mathcal{F}_i$, $i = 1 \ldots n$, meets the criterion of [Gumm & Schröder(2002)], also meets the criterion itself and hence $\mathcal{F}_1 \times \cdots \times \mathcal{F}_n$ is bounded.

For bounded functors we have the following result, see [Gumm & Schröder(2001)] for a proof.

Theorem 3 *If a functor $\mathcal{F}$:* **Set** *$\to$* **Set** *is bounded, then its final coalgebra exists.* □

An $\mathcal{F}$-coalgebra (R, γ) with $R \subseteq X \times Y$ is called a coalgebraic bisimulation of two $\mathcal{F}$-coalgebras (X, α) and (Y, β) if $\alpha \circ \pi_1 = \mathcal{F}(\pi_1) \circ \gamma$ and $\beta \circ \pi_2 = \mathcal{F}(\pi_2) \circ \gamma$. Here $\pi_1 : R \to X$ and $\pi_2 : R \to Y$ are the projections from R to X and Y, respectively. See Figure 1. For a coalgebra $\mathcal{S} = (X, \alpha)$, two elements $x_1, x_2 \in X$ are called coalgebraically bisimilar if there exists a coalgebraic bisimulation R of (X, α) and (X, α) such that $R(x_1, x_2)$, notation $x_1 \sim^{\mathcal{S}}_{\mathcal{F}} x_2$, or simply $x_1 \sim_{\mathcal{F}} x_2$ when the coalgebra $\mathcal{S}$ is clear from the context.

Given a cospan $f_1 : A \to C$ and $f_2 : B \to C$, their pullback in **Set** is a set P together with a span $p_1 : P \to A$ and $p_2 : P \to B$ such that $f_1 \circ p_1 = f_2 \circ p_2$, and, moreover, for any Q and span $q_1 : Q \to A$ and $q_2 : Q \to B$ satisfying $f_1 \circ q_1 = f_2 \circ q_2$ there exists a unique mapping $m : Q \to P$, called the mediating morphism, such that $q_1 = p_1 \circ m$ and $q_2 = p_2 \circ m$. In case the mediating morphism need not to be unique, we speak of a weak pullback of $f_1 : A \to C$ and $f_2 : B \to C$. In **Set**, the pullback of $f_1 : A \to C$ and $f_2 : B \to C$ is the set $P = \{ (a, b) \in A \times B \mid f_1(a) = f_2(b) \}$ together with the projections $\pi_1 : P \to A$ and $\pi_2 : P \to B$.

If a functor $\mathcal{F}$ transforms a weak pullback diagram for $f : A \to C$ and $g : B \to C$ into a weak pullback diagram of $\mathcal{F}(f) : \mathcal{F}(A) \to \mathcal{F}(C)$, and $\mathcal{F}(g) : \mathcal{F}(B) \to \mathcal{F}(C)$, the functor $\mathcal{F}$ is said to preserve weak pullbacks. More explicitly, $\mathcal{F}$ preserves weak pullbacks, if for any cospan $f : A \to C$ and $g : B \to C$ in **Set**, having a weak pullback W together with span w_1 and w_2, we have that $\mathcal{F}(f) \circ \mathcal{F}(w_1) = \mathcal{F}(f) \circ \mathcal{F}(w_2)$, and, for any set Q and span $F : Q \to \mathcal{F}(A)$ and $G : Q \to \mathcal{F}(B)$ with $\mathcal{F}(f) \circ F = \mathcal{F}(g) \circ G$, we can find a mediating morphism $M : Q \to \mathcal{F}(W)$ such that $F = \mathcal{F}(w_1) \circ M$ and $G = \mathcal{F}(w_2) \circ M$. See Figure 2.

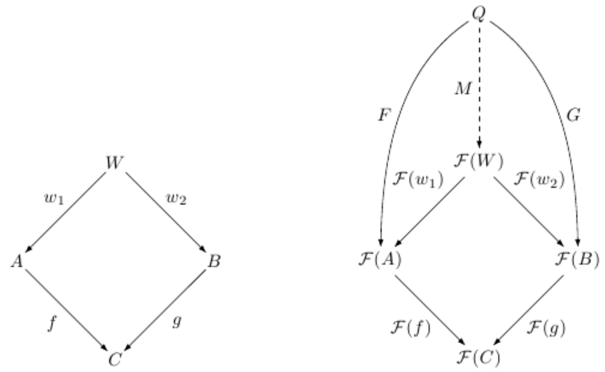

Figure 2: Functor $\mathcal{F}$ preserving weak pullbacks

In **Set**, for many functors coalgebraic bisimulation and behavioural equivalence coincide (but not always, see [Bonchi et al.(2011), Section 2.2] for an example). A sufficient condition for the two notions being equal is the preservation of weak pullbacks, which is for many functors the case (but not all, in particular not for the Giry-functor on the category of measurable spaces and functions, see [Viglizzo(2005), Marti & Venema(2012)]).

Theorem 4 *[Rutten(2000), Theorem 9.3] If a functor $\mathcal{F}$ on* **Set** *preserves weak pullbacks, then coalgebraic bisimilarity and behavioural equivalence coincide, i.e. for any $\mathcal{F}$-coalgebra $\mathcal{S} = (X, \alpha)$ and elements $x_1, x_2 \in X$ it holds that $x_1 \sim^{\mathcal{S}}_{\mathcal{F}} x_2$ iff $x_1 \approx^{\mathcal{S}}_{\mathcal{F}} x_2$.* □

A number of proofs of results on process languages $\mathcal{P}$ in this paper rely on so-called guarded recursion [de Bakker & de Vink(1996)]. Typically, constants X, also called process variables, 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, i.e. 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 $c(P) = 1$ if P is a prefix construct, $c(P_1 \bullet P_2) > \max\{\, c(P_1),\, c(P_2) \,\}$ for all other syntactic operators $\bullet$ of $\mathcal{P}$, and moreover $c(X) > c(P)$ if $X := P$.

3 State-to-Function Labelled Transition Systems

The definition of a state-to-function labelled transition system, *FuTS* for short, involves a set S of states and one or more relations of states on the one hand, and functions from states into semirings on the other hand. 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 state-to-function labelled 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, for $i = 1 \ldots n$, $\rightarrowtail_i \subseteq S \times \mathcal{L}_i \times \mathcal{FS}(S, \mathcal{R}_i)$.* •

Similar as for state-to-state transitions of *LTS*s, for state-to-function transitions of *FuTS*s 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 or the carrier set. We refer to each $\rightarrowtail_i$ as a state-to-function transition relation of $\mathcal{S}$ or just as a transition relation. If for $\mathcal{S}$ we have that $n = 1$, i.e. there is only one state-to-function transition relation $\rightarrowtail$,

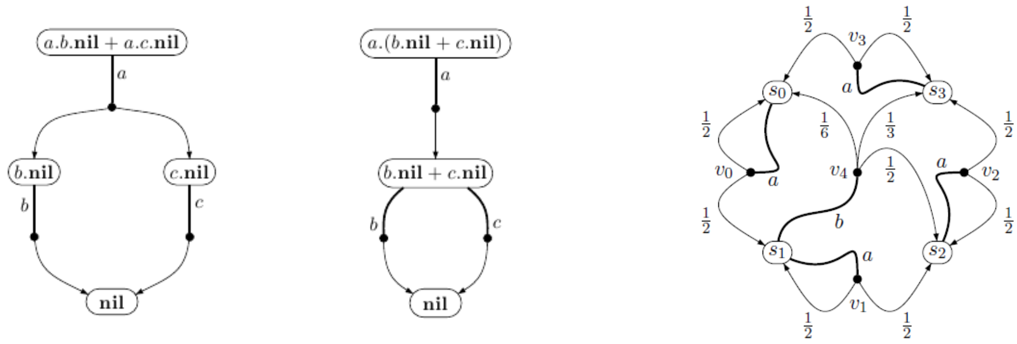

Figure 3: *FuTS* for two standard processes and a probabilistic process.

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 state-to-function relations $\rightarrowtail_i$ correspond to functions $S \to \mathcal{L}_i \to \mathcal{FS}(S, \mathcal{R}_i)$. For the remainder of the paper, all *FuTS*s we consider will be total and deterministic, unless explicitly stated otherwise. It is noted that Definition 1 slightly differs in formulation from the one provided in [De Nicola et al.(2011)].

Examples For the modeling of standard interactive processes with *FuTS*s, we choose a set of actions $\mathcal{A}$ as label set and the booleans $\mathbb{B}$ as semiring. Consider the two processes $P = a.b.\mathbf{nil} + a.c.\mathbf{nil}$ and $Q = a.(b.\mathbf{nil} + c.\mathbf{nil})$, their representation as a *FuTS* is depicted in Figure 3. For process P, on the left of the figure, we have $P \overset{a}{\rightarrowtail} [b.\mathbf{nil} \mapsto \mathtt{true},\ c.\mathbf{nil} \mapsto \mathtt{true}]$, while for the process Q, in the middle of the figure, we have $Q \overset{a}{\rightarrowtail} [b.\mathbf{nil} + c.\mathbf{nil} \mapsto \mathtt{true}]$. So, for P the two processes $b.\mathbf{nil}$ and $c.\mathbf{nil}$ each are set to $\mathtt{true}$ and for Q only the process $b.\mathbf{nil} + c.\mathbf{nil}$ is set to $\mathtt{true}$. Since any finite number of alternatives can be assigned a non-zero value by a function of finite support, deterministic *FuTS*s are able to represent image-finite non-determinism; the branching is taken care of by the functions from process terms to $\mathbb{B}$. To complete the picture $b.\mathbf{nil} \overset{b}{\rightarrowtail} [\mathbf{nil} \mapsto \mathtt{true}]$, $c.\mathbf{nil} \overset{c}{\rightarrowtail} [\mathbf{nil} \mapsto \mathtt{true}]$, $b.\mathbf{nil} + c.\mathbf{nil} \overset{b}{\rightarrowtail} [\mathbf{nil} \mapsto \mathtt{true}]$ and $b.\mathbf{nil} + c.\mathbf{nil} \overset{c}{\rightarrowtail} [\mathbf{nil} \mapsto \mathtt{true}]$.

As another example of a simple *FuTS*, Figure 3 displays at its right 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_3 used in the example have the property that $\oplus v_i(s) = 1$, for $i = 0 \ldots 3$. More explicitly we have

$$s_0 \overset{a}{\rightarrowtail} [s_0 \mapsto \tfrac{1}{2},\ s_1 \mapsto \tfrac{1}{2}] \qquad s_2 \overset{a}{\rightarrowtail} [s_2 \mapsto \tfrac{1}{2},\ s_3 \mapsto \tfrac{1}{2}] \qquad s_3 \overset{a}{\rightarrowtail} [s_0 \mapsto \tfrac{1}{2},\ s_3 \mapsto \tfrac{1}{2}]$$

$$s_1 \overset{a}{\rightarrowtail} [s_1 \mapsto \tfrac{1}{2},\ s_2 \mapsto \tfrac{1}{2}] \qquad s_1 \overset{b}{\rightarrowtail} [s_0 \mapsto \tfrac{1}{6},\ s_2 \mapsto \tfrac{1}{2},\ s_3 \mapsto \tfrac{1}{3}]$$

$$s_i \overset{b}{\rightarrowtail} [\,]_{\mathbb{B}} \ \text{ for } i = 0, 2, 3$$

Usually, such a *FuTS* over $\mathbb{R}_{\geqslant 0}$ is called a (reactive) probabilistic transition system, using standard terminology introduced in [van Glabbeek et al.(1995)].

In Section 7 we will provide semantics for the process language *IML* for interactive Markov chains [Hermanns(2002), Hermanns & Katoen(2010)] using *FuTS*s. 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.

Below 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 frequently for concrete examples in the literature. Alternatively, using disjoint unions, one could see a *FuTS* represented by a function θ' of type $S \to \bigoplus_{i=1}^n \mathcal{L}_i \to \bigoplus_{i=1}^n \mathcal{FS}(S, \mathcal{R}_i)$ 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.

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 $\ell \in \mathcal{L}_i$, where we use the notation $[t]_R$ to denote the equivalence class of $t \in S$ with respect to R. 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 $x_1 \simeq_{\mathcal{S}} x_2$. •

Note that the sums in equation (1) exist since the functions $\theta_i(s_1)(\ell), \theta_i(s_2)(\ell) \in \mathcal{FS}(S, \mathcal{R}_i)$, $i = 1 \ldots n$, are of finite support.

For the combined *FuTS* of the two processes $P = a.b.\mathbf{nil} + a.c.\mathbf{nil}$ and $Q = a.(b.\mathbf{nil} + c.\mathbf{nil})$ of Figure 3, consider the equivalence relation R such that $R(P, Q)$ and also $R(b.\mathbf{nil}, b.\mathbf{nil} + c.\mathbf{nil})$, $R(b.\mathbf{nil}, b.\mathbf{nil} + c.\mathbf{nil})$, $R(c.\mathbf{nil}\, b.\mathbf{nil} + c.\mathbf{nil})$, and $R(\mathbf{nil}, \mathbf{nil})$. Then R is not a *FuTS*-bisimulation. Although, on the one hand, $\sum_{t' \in [\mathbf{nil}]_R} \theta(b.\mathbf{nil})(b)(t') = \theta(b.\mathbf{nil})(b)(\mathbf{nil}) = \mathtt{true}$ and, on the other hand, $\sum_{t' \in [\mathbf{nil}]_R} \theta(b.\mathbf{nil} + c.\mathbf{nil})(b)(t') = \theta(b.\mathbf{nil} + c.\mathbf{nil})(b)(\mathbf{nil}) = \mathtt{true}$, we have $\sum_{t' \in [\mathbf{nil}]_R} \theta(b.\mathbf{nil})(c)(t') = \mathtt{false}$, while $\sum_{t' \in [\mathbf{nil}]_R} \theta(b.\mathbf{nil} + c.\mathbf{nil})(c)(t') = \mathtt{true}$, taking sums, i.e. disjunctions, in $\mathbb{B}$. As no other equivalence relation fulfills the requirements of Definition 2 either, we conclude that the processes P and Q are not bisimilar. See Section 5 for more detail.

For *FuTS*s that are neither total nor deterministic a variation of Definition 2 applies involving the usual transfer conditions: if $R(s_1, s_2)$ then

$$\begin{array}{l} s_1 \overset{\ell}{\rightarrowtail}_i v_1 \in \mathcal{FS}(S, \mathcal{R}_i) \Longrightarrow \exists v_2 \colon s_2 \overset{\ell}{\rightarrowtail}_i v_2 \in \mathcal{FS}(S, \mathcal{R}_i) \wedge \tilde{R}(v_1, v_2) \\ s_2 \overset{\ell}{\rightarrowtail}_i v_2 \in \mathcal{FS}(S, \mathcal{R}_i) \Longrightarrow \exists v_1 \colon s_1 \overset{\ell}{\rightarrowtail}_i v_1 \in \mathcal{FS}(S, \mathcal{R}_i) \wedge \tilde{R}(v_1, v_2) \end{array}$$

where the lifting $\tilde{R}$ on $\mathcal{FS}(S, \mathcal{R}_i) \times \mathcal{FS}(S, \mathcal{R}_i)$ is given by

$$\tilde{R}(v_1, v_2) \iff \textstyle\sum_{t' \in [t]_R} v_1(t') = \sum_{t' \in [t]_R} v_2(t')$$

As we will consider total and deterministic *FuTS*s only, we stick to the more convenient formulation involving transition functions of Definition 2.

4 *FuTS*s coalgebraically

In this section we will cast *FuTS*s in the framework of coalgebras and prove a correspondence result of *FuTS*-bisimulation and behavioural equivalence for a suitable functor on **Set**. We also show that for the functors associated with *FuTS*s, under mild conditions for the semirings involved, behavioural equivalence and coalgebraic bisimulation coincide.

Definition 3 *Let $\mathcal{L}$ be a set of labels and $\mathcal{R}$ a semiring. The functor $\mathcal{U}^{\mathcal{L}}_{\mathcal{R}} : \mathbf{Set} \to \mathbf{Set}$ 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 mapping $f : X \to Y$ the mapping $\mathcal{U}^{\mathcal{L}}_{\mathcal{R}}(f) : \mathcal{FS}(X,\mathcal{R})^{\mathcal{L}} \to \mathcal{FS}(Y,\mathcal{R})^{\mathcal{L}}$ where*

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

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 $\mathcal{U}^{\mathcal{L}}_{\mathcal{R}}$ to be well-defined. In fact, we have $spt(\,\mathcal{U}^{\mathcal{L}}_{\mathcal{R}}(f)(\varphi)(\ell)\,) \subseteq \{\, f(x) \mid x \in spt(\varphi)(\ell)\,\}$. We furthermore observe that for any simple *FuTS* (S,θ) over $\mathcal{L}$ and $\mathcal{R}$ we have $\theta : S \to \mathcal{L} \to \mathcal{FS}(S,\mathcal{R})$. Thus (S,θ) can be interpreted as a $\mathcal{U}^{\mathcal{L}}_{\mathcal{R}}$-coalgebra. In the sequel, we will abbreviate $\mathcal{U}^{\mathcal{L}}_{\mathcal{R}}$ with $\mathcal{U}$ whenever $\mathcal{L}$ and $\mathcal{R}$ are clear from the context.

As we aim to compare our notion of bisimulation for *FuTS*s with behavioural equivalence for the functor $\mathcal{U}$, given a set of labels $\mathcal{L}$ and a semiring $\mathcal{R}$, we need to check that $\mathcal{U}$ possesses a final coalgebra. For this, we adapt the proof for the functor $\mathcal{FS}(\cdot,\mathcal{M}) : \mathbf{Set} \to \mathbf{Set}$ where $\mathcal{M}$ is a monoid (rather than a semiring) as sketched in [Silva(2010)] to the setting here. The proof exploits the characterisation of [Gumm & Schröder(2002)], see Section 2. An alternative route to showing the existence of a final coalgebra is followed in [Klin(2009)], also for commutative monoids, and relies on a finitary result of Barr, cf. [Barr(1993)].

Lemma 5 *Let $\mathcal{L}$ be a set of labels, $\mathcal{R}$ a semiring. Then the functor $\mathcal{U}$ on $\mathbf{Set}$ is bounded.*

Proof We verify that $\eta : \mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}} \times (\cdot)^{\mathcal{L}\times\mathbb{N}} \Rightarrow \mathcal{FS}(\cdot,\mathcal{R})^{\mathcal{L}}$, defined by $\eta_X : \mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}} \times X^{\mathcal{L}\times\mathbb{N}} \to \mathcal{FS}(X,\mathcal{R})^{\mathcal{L}}$ such that $\eta_X(\nu,\xi)(\ell)(x) = \sum_{(\ell,n)\in\xi^{-1}(x)} \nu(\ell)(n)$, is a natural transformation with surjective components. Note that η is of the form $A \times (\cdot)^B$ for the sets $A = \mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}$ and $B = \mathcal{L} \times \mathbb{N}$ as the characterisation requires.

For $f : X \to Y$ we check that the diagram

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

commutes, where $f^{\mathcal{L}\times\mathbb{N}} : X^{\mathcal{L}\times\mathbb{N}} \to Y^{\mathcal{L}\times\mathbb{N}}$ is, as usual, given by $f^{\mathcal{L}\times\mathbb{N}}(\xi) = f \circ \xi$. We have, for $\nu \in \mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}$, $\xi \in X^{\mathcal{L}\times\mathbb{N}}$, $\ell \in \mathcal{L}$ and $y \in Y$, that

$$\begin{array}{lll}
\multicolumn{2}{l}{\mathcal{FS}(f,\mathcal{R})^{\mathcal{L}}(\,\eta_X(\nu,\xi)\,)(\ell)(y)} & \\
= & \sum_{x \in f^{-1}(y)} \eta_X(\nu,\xi)(\ell)(x) & \text{definition } \mathcal{FS}(f,\mathcal{R})^{\mathcal{L}} \\
= & \sum_{x \in f^{-1}(y)} \sum_{(\ell,n)\in\xi^{-1}(x)} \nu(\ell)(n) & \text{definition } \eta_X \\
= & \sum_{(\ell,n)\in(f\circ\xi)^{-1}(y)} \nu(\ell)(n) & (f\circ\xi)^{-1}(y) = \xi^{-1}(f^{-1}(y)) \\
= & \eta_Y(\,\nu, f\circ\xi\,)(\ell)(y) & \text{definition } \eta_Y \\
= & \eta_Y((\,id_{\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}} \times f^{\mathcal{L}\times\mathbb{N}})(\nu,\xi)\,)(\ell)(y) & \text{definitions of } id_{\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}} \text{ and } f^{\mathcal{L}\times\mathbb{N}}
\end{array}$$

Thus, we have that $\mathcal{FS}(f,\mathcal{R})^{\mathcal{L}}(\,\eta_X(v,\xi)\,) = \eta_Y((\,id_{\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}} \times f^{\mathcal{L}\times\mathbb{N}})(v,\xi)\,)$ and $\mathcal{FS}(f,\mathcal{R})^{\mathcal{L}} \circ \eta_X = \eta_Y \circ (\,id_{\mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}} \times f^{\mathcal{L}\times\mathbb{N}}\,)$.

As to the surjectivity of $\eta_X : \mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}} \times X^{\mathcal{L}\times\mathbb{N}} \to \mathcal{FS}(X,\mathcal{R})^{\mathcal{L}}$, for any set X: Choose $\varphi \in \mathcal{FS}(X,\mathcal{R})^{\mathcal{L}}$. Pick $\ell \in \mathcal{L}$. Suppose $spt(\varphi(\ell)) = \{x_1,\dots,x_s\} \subseteq X$. Pick $x_0 \in X$ arbitrary. We define $v_\varphi \in \mathcal{FS}(\mathbb{N},\mathcal{R})^{\mathcal{L}}$ and $\xi : \mathcal{L}\times\mathbb{N} \to X$ as follows:

$$v_\varphi(\ell)(i) = \begin{cases} \varphi(\ell)(x_i) & \text{for } i = 1\dots s \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \xi_\varphi(\ell,i) = \begin{cases} x_i & \text{for } i = 1\dots s \\ x_0 & \text{otherwise} \end{cases}$$

Note, $v_\varphi(\ell)$ has finite support, for $\ell \in \mathcal{L}$. Then we have

$$\begin{array}{lll} & \eta_X(v_\varphi,\xi_\varphi)(\ell)(x) & \\ = & \sum_{(\ell,n)\,\in\,\xi_\varphi^{-1}(x)} v_\varphi(\ell)(n) & \text{definition } \eta_X \\ = & \sum_{(\ell,i)\,\in\,\xi_\varphi^{-1}(\ell,x),\ 1\leqslant i\leqslant s} v_\varphi(\ell)(i) & v_\varphi(\ell)(n) = 0 \text{ for } n \neq 1\dots s \\ = & \begin{cases} \varphi(\ell)(x_i) & \text{if } x = x_i,\ i = 1\dots s \\ 0 & \text{otherwise} \end{cases} & \text{sums of 0 or 1 summand} \\ = & \varphi(\ell)(x) & spt(\varphi)(\ell) = \{x_1,\dots,x_s\} \end{array}$$

Hence $\eta_X(v_\varphi,\xi_\varphi) = \varphi$ and η_X is surjective.

Working with total and deterministic *FuTS*s, we can interpret a *FuTS* $\mathcal{S} = (\,S,\,\langle\,\theta_i\,\rangle_{i=1}^n\,)$ over the label sets $\mathcal{L}_i$ and semirings $\mathcal{R}_i$, $i = 1\dots n$, 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 $L = \langle \mathcal{L}_1,\dots,\mathcal{L}_n \rangle$ be an n-tuple of label sets and $R = \langle \mathcal{R}_1,\dots,\mathcal{R}_n \rangle$ be an n-tuple of semirings. The functor $\mathcal{V}_R^L$ on* **Set** *is defined by $\mathcal{V}_R^L = \prod_{i=1}^n \mathcal{U}_{\mathcal{R}_i}^{\mathcal{L}_i} = \prod_{i=1}^n \mathcal{FS}(\,\cdot\,,\mathcal{R}_i\,)^{\mathcal{L}_i}$.*

We note that any *FuTS* $\mathcal{S} = (\,S,\,\langle\,\theta_i\,\rangle_{i=1}^n\,)$ over label sets $\mathcal{L}_i$ and semirings $\mathcal{R}_i$, for $i = 1\dots n$, is a $\mathcal{V}_R^L$-coalgebra. In the sequel, we shall use $\mathcal{V}$ as an abbreviation for $\mathcal{V}_R^L$ whenever $L = \langle \mathcal{L}_1,\dots,\mathcal{L}_n \rangle$ and $R = \langle \mathcal{R}_1,\dots,\mathcal{R}_n \rangle$ are clear from the context. Similarly, and for the sake of readability, we shall often abbreviate $\mathcal{U}_{\mathcal{R}_i}^{\mathcal{L}_i}$ by $\mathcal{U}_i$.

Under conditions that are generally met, coalgebras come equipped with a natural notion of behavioural equivalence that can act as a reference for strong equivalences, in particular of bisimulation for *FuTS*s. Below, see Theorem 7, we prove that $\mathcal{S}$-bisimilarity as given by Definition 2 coincides with behavioural equivalence for the functor $\mathcal{V}$ as given by Definition 4, providing justification for the notion of equivalence defined on *FuTS*s.

For the notion of behavioural equivalence for the functor $\mathcal{V}$ to be defined, we need to verify that it possesses a final coalgebra.

Theorem 6 *The functor $\mathcal{V}$ has a final coalgebra.*

Proof From the proof of Lemma 5 we obtain that, for each factor $\mathcal{U}_i$ of $\mathcal{V}$, there exist sets A_i and B_i and a surjective natural transformation $\eta^i : A_i \times (\cdot)^{B_i} \Rightarrow \mathcal{U}_i$, $i = 1\dots n$. By Lemma 2 it follows that we can find sets A and B and a surjective natural transformation $\eta : A \times (\cdot)^B \Rightarrow \mathcal{V}$. Hence, by Theorem 3, it follows that the functor $\mathcal{V}$ possesses a final coalgebra (Ω,ω).

Since the functor $\mathcal{V}$ has a final coalgebra, we can speak of the behavioural equivalence $\approx_{\mathcal{V}}$ on any $\mathcal{V}$-coalgebra or, equivalently, of the *FuTS* $\mathcal{S}$. Writing $[\![\cdot]\!]_{\mathcal{V}}$ for the final morphism of a $\mathcal{V}$-coalgebra $\mathcal{S}$ into (Ω, ω), we have

$$[\![\cdot]\!]_{\mathcal{V}} = [\![\cdot]\!]_{\mathcal{U}_1} \times \cdots \times [\![\cdot]\!]_{\mathcal{U}_n}$$

Next we establish, for a given *FuTS* $\mathcal{S}$ over $\mathcal{L}_1, \ldots, \mathcal{L}_n$ and $\mathcal{R}_1 \ldots, \mathcal{R}_n$ the correspondence of $\mathcal{S}$-bisimulation $\simeq_{\mathcal{S}}$ as given by Definition 2 and the behavioural equivalence $\approx_{\mathcal{V}}$.

Theorem 7 *Let $\mathcal{S} = (\, S, \langle\, \theta_i \,\rangle_{i=1}^n \,)$ be a FuTS over the label sets $\mathcal{L}_i$ and semirings $\mathcal{R}_i$, $i = 1 \ldots n$, and $\mathcal{V}$ as in Definition 4. Then $s_1 \simeq_{\mathcal{S}} s_2 \Leftrightarrow s_1 \approx_{\mathcal{V}} s_2$, for all $s_1, s_2 \in S$.*

Proof Let $s_1, s_2 \in S$. We first prove $s_1 \simeq_{\mathcal{S}} s_2 \;\Rightarrow\; s_1 \approx_{\mathcal{V}} s_2$. So, assume $s_1 \simeq_{\mathcal{S}} s_2$. 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 $\mathcal{V}$-coalgebra. We turn the collection of equivalence classes S/R into a $\mathcal{V}$-coalgebra $\mathcal{S}_R = (S/R, \varrho_R)$ by putting $\varrho_R = \varrho_1 \times \cdots \times \varrho_n$ where

$$\varrho_i(\, [s]_R \,)(\ell)(\, [t]_R \,) = \textstyle\sum_{t' \in [t]_R} \theta_i(s)(\ell)(t')$$

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 $\mathcal{V}$-homomorphism: For $i = 1 \ldots n$, $\ell \in \mathcal{L}_i$ and $t \in S$, we have

$$\begin{array}{lll}
\multicolumn{3}{l}{\mathcal{U}_i(\varepsilon_R)(\, \theta_i(s) \,)(\ell)([t]_R)} \\
= & \sum_{\, t' \in \varepsilon_R^{-1}([t]_R)} \theta_i(s)(\ell)(t') & \text{by definition of } \mathcal{U}_i \\
= & \sum_{\, t' \in [t]_R} \theta_i(s)(\ell)(t') & \text{by definition of } \varepsilon_R \\
= & \varrho_i(\, [s]_R \,)(\ell)(\, [t]_R \,) & \text{by definition of } \varrho_i \\
= & \varrho_i(\, \varepsilon_R(s) \,)(\ell)(\, [t]_R \,) & \text{by definition of } \varepsilon_R
\end{array}$$

Thus, $\mathcal{U}_i(\varepsilon_R) \circ \theta_i = \varrho_i \circ \varepsilon_R$. Since $\mathcal{V}(\varepsilon_R) = \prod_{i=1}^n \mathcal{U}_i(\varepsilon_R)$ it follows that ε_R is a $\mathcal{V}$-homomorphism. Therefore, by uniqueness of a final morphism, we have $[\![\cdot]\!]_{\mathcal{V}}^{\mathcal{S}} = [\![\cdot]\!]_{\mathcal{V}}^{\mathcal{S}_R} \circ \varepsilon_R$. In particular, with respect to $\mathcal{S}$, this implies $[\![s_1]\!]_{\mathcal{V}} = [\![s_2]\!]_{\mathcal{V}}$ since $\varepsilon_R(s_1) = \varepsilon_R(s_2)$. Thus, $s_1 \approx_{\mathcal{V}} s_2$.

For the reverse, $s_1 \approx_{\mathcal{V}} s_2 \;\Rightarrow\; s_1 \simeq_{\mathcal{S}} s_2$, assume $s_1 \approx_{\mathcal{V}} s_2$, i.e. $[\![s_1]\!]_{\mathcal{V}} = [\![s_2]\!]_{\mathcal{V}}$, for $s_1, s_2 \in S$. Since the map $[\![\cdot]\!]_{\mathcal{V}} : (S, \theta) \to (\Omega, \omega)$ is a $\mathcal{V}$-homomorphism, the equivalence relation $R_{\mathcal{S}}$ with $R_{\mathcal{S}}(s', s'') \Leftrightarrow [\![s']\!]_{\mathcal{V}} = [\![s'']\!]_{\mathcal{V}}$ is an $\mathcal{S}$-bisimulation: Suppose $R_{\mathcal{S}}(s', s'')$, i.e. $s' \approx_{\mathcal{V}} s''$, for some $s', s'' \in S$. Assume $\omega = \omega_1 \times \cdots \times \omega_n$. Pick $1 \leqslant i \leqslant n$, $\ell \in \mathcal{L}_i$, $t \in S$ and assume $[\![t]\!]_{\mathcal{V}} = w \in \Omega$. Since $[\![\cdot]\!]_{\mathcal{V}} : (S, \theta) \to (\Omega, \omega)$ is a $\mathcal{V}$-homomorphism we have, for $i = 1 \ldots n$, that $\omega_i \circ [\![\cdot]\!]_{\mathcal{V}} = \mathcal{U}_i([\![\cdot]\!]_{\mathcal{V}}) \circ \theta_i$. Hence, for $s \in S$, it holds that

$$\omega_i(\, [\![s]\!]_{\mathcal{V}} \,)(\ell)(w) = \mathcal{U}_i([\![\cdot]\!]_{\mathcal{V}})(\theta_i(s))(\ell)(w) = \textstyle\sum_{\, t' \in [\![\cdot]\!]_{\mathcal{V}}^{-1}(w)} \theta_i(s)(\ell)(t') \qquad (2)$$

Therefore we have

$$\begin{array}{lll}
\multicolumn{3}{l}{\sum_{\, t' \in [t]_{R_{\mathcal{S}}}} \theta_i(s')(\ell)(t')} \\
= & \sum_{\, t' \in [\![\cdot]\!]_{\mathcal{V}}^{-1}(w)} \theta_i(s')(\ell)(t') & \text{by definition of } R_{\mathcal{S}} \text{ and } w \\
= & \omega_i(\, [\![s']\!]_{\mathcal{V}} \,)(\ell)(w) & \text{by equation (2)} \\
= & \omega_i(\, [\![s'']\!]_{\mathcal{V}} \,)(\ell)(w) & s' \approx_{\mathcal{V}} s'' \text{ by assumption} \\
= & \sum_{\, t' \in [\![\cdot]\!]_{\mathcal{V}}^{-1}(w)} \theta_i(s'')(\ell)(t') & \text{by equation (2)} \\
= & \sum_{\, t' \in [t]_{R_{\mathcal{S}}}} \theta_i(s'')(\ell)(t') & \text{by definition of } R_{\mathcal{S}} \text{ and } w
\end{array}$$

Thus, if $R_{\mathcal{S}}(s', s'')$ then $\sum_{t' \in [t]_{R_{\mathcal{S}}}} \theta_i(s')(\ell)(t') = \sum_{t' \in [t]_{R_{\mathcal{S}}}} \theta_i(s'')(\ell)(t')$ for all $t \in S$, $i = 1 \ldots n$, $\ell \in \mathcal{L}_i$, and therefore $R_{\mathcal{S}}$ is an $\mathcal{S}$-bisimulation. Since $[\![s_1]\!]_{\mathcal{V}} = [\![s_2]\!]_{\mathcal{V}}$, 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 $s_1 \simeq_{\mathcal{S}} s_2$.

We continue with relating *FuTS* bisimilarity via behavioural equivalence to coalgebraic bisimulation. We will show that, for a *FuTS* $\mathcal{S}$ over $L = (\mathcal{L}_1, \ldots, \mathcal{L}_n)$ and $R = (\mathcal{R}_1, \ldots, \mathcal{R}_n)$, when seen as a coalgebra, behavioural equivalence $\approx_{\mathcal{V}}$ and coalgebraic bisimulation $\sim_{\mathcal{V}}$ coincide. Thus in view of Theorem 7, we will have that *FuTS* bisimilarity $\simeq_{\mathcal{S}}$ and coalgebraic bisimulation $\sim_{\mathcal{V}}$ are the same. For this, it suffices by Theorem 4 to verify that the functor $\mathcal{V}$ preserves weak pullbacks. However, our construction below requires that the semirings involved satisfy two additional requirements: (i) existence of a (right) multiplicative inverse for non-zero elements, i.e. for $r \in \mathcal{R} \backslash \{0_{\mathcal{R}}\}$ it holds that $r *_{\mathcal{R}} r' = 1_{\mathcal{R}}$ for some $r' \in \mathcal{R}$; (ii) the *zero-sum property*, stating that a sum $r_1 +_{\mathcal{R}} \cdots +_{\mathcal{R}} r_n = 0_{\mathcal{R}}$ iff each $r_i = 0_{\mathcal{R}}$, $i = 1 \ldots n$. Thus, first, non-degenerate quotients exist, and, second, non-zero elements cannot be canceled out. In the concrete case of $\mathbb{B}$ and $\mathbb{R}_{\geqslant 0}$, that will be used in the sequel, these requirements are clearly fulfilled. For readability, we will use standard notation like r_1/r_2 rather than $r_1 *_{\mathcal{R}} r_2^{-1}$.

We first consider the case of a simple *FuTS* and establish the preservation of weak pullbacks for the functor $\mathcal{U}$ of Definition 3, moving to the functor $\mathcal{V}$ of Definition 4, of a general *FuTS* afterward.

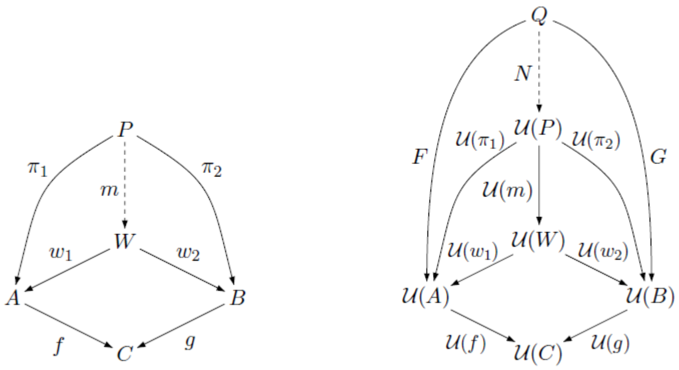

Figure 4: Functor $\mathcal{U}$ preserves weak pullbacks

Lemma 8 *For a set of labels $\mathcal{L}$ and a semiring $\mathcal{R}$, with multiplicative inverse and satisfying the zero-sum property, the functor $\mathcal{U}$ of Definition 3 preserves weak pullbacks.*

Proof See Figure 4. Let $f : A \to C$ and $g : B \to C$ be a cospan. Put $P = \{ (a,b) \in A \times B \mid f(a) = g(b) \}$ and let $\pi_1 : P \to A$ and $\pi_2 : P \to B$ be the projections. Then P is the pullback of f and g. Clearly, $\mathcal{U}(f) \circ \mathcal{U}(\pi_1) = \mathcal{U}(g) \circ \mathcal{U}(\pi_2)$ as $f \circ \pi_1 = g \circ \pi_2$.

Suppose $F : Q \to \mathcal{U}(A)$ and $G : Q \to \mathcal{U}(B)$ are such that $\mathcal{U}(f) \circ F = \mathcal{U}(g) \circ G$. Then we define $N : Q \to \mathcal{U}(P)$ by

$$N(q)(\ell)(a,b) = \begin{cases} \dfrac{F(q)(\ell)(a) * G(q)(\ell)(b)}{\mathcal{U}(g)(G(q))(\ell)(f(a))} & \text{if } \mathcal{U}(g)(G(q))(\ell)(f(a)) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

for $\ell \in \mathcal{L}$ and $(a,b) \in P$. Note that the above construction is well-defined as $\mathcal{R}$ is assumed to have quotients. We claim that

$$F = \mathcal{U}(\pi_1) \circ N \text{ and } G = \mathcal{U}(\pi_2) \circ N \tag{3}$$

Note, for $\ell \in \mathcal{L}$, $(a, b) \in P$ and $q \in Q$, we have

$$\mathcal{U}(g)(G(q))(\ell)(f(a)) = \mathcal{U}(g)(G(q))(\ell)(g(b)) = \mathcal{U}(f)(F(q))(\ell)(g(b)) \tag{4}$$

since $f(a) = g(b)$ and $\mathcal{U}(f) \circ F = \mathcal{U}(g) \circ G$. So, the definition of N is symmetric in a and b.

We verify that $F = \mathcal{U}(\pi_1) \circ N$. Pick $q \in Q$, $\ell \in \mathcal{L}$ and $a \in A$. We must show $F(q)(\ell)(a) = \mathcal{U}(\pi_1)(N(q))(\ell)(a)$. We distinguish two cases.
Case I: $\mathcal{U}(g)(G(q))(\ell)(f(a)) = 0$. Then we have

$$\mathcal{U}(\pi_1)(N(q))(\ell)(a) = \textstyle\sum \{ N(q)(\ell)(a, b) \mid b \in B \colon f(a) = g(b) \} = 0$$

by definition of N.
On the other hand, if we have $\mathcal{U}(g)(G(q))(\ell)(f(a)) = 0$, then also have $\mathcal{U}(f)(F(q))(\ell)(f(a)) = 0$ since $\mathcal{U}(f) \circ F = \mathcal{U}(g) \circ G$. Since

$$\mathcal{U}(f)(\, F(q)\,)(\ell)(f(a)) = \textstyle\sum_{a' \in f^{-1}(f(a))} F(q)(\ell)(a') = \sum_{a' : f(a') = f(a)} F(q)(\ell)(a')$$

we obtain $\sum \{ F(q)(\ell)(a') \mid a \in A \colon f(a') = f(a) \} = 0$. In particular, by the zero-sum property of $\mathcal{R}$, $F(q)(\ell)(a) = 0$. Hence, both $F(q)(\ell)(a) = 0$ and $\mathcal{U}(\pi_1)(N(q))(\ell)(a) = 0$ and it follows that $F(q)(\ell)(a) = \mathcal{U}(\pi_1)(N(q))(\ell)(a)$.

Case II: $\mathcal{U}(g)(G(q))(\ell)(f(a)) \neq 0$. Then we have, since $(a, b) \in \pi_1^{-1}(a)$ iff $f(a) = g(b)$,

$$\begin{array}{lll}
\multicolumn{2}{l}{\mathcal{U}(\pi_1)(N(q))(\ell)(a)} & \\
= & \sum_{b : f(a) = g(b)} N(q)(\ell)(a, b) & \text{by definition of } \mathcal{U}(\pi_1) \\
= & \sum_{b : f(a) = g(b)} \dfrac{F(q)(\ell)(a) * G(q)(\ell)(b)}{\mathcal{U}(g)(G(q))(\ell)(f(a))} & \text{by definition of } N \\
= & F(q)(\ell)(a) * \dfrac{\sum_{b : f(a) = g(b)} G(q)(\ell)(b)}{\mathcal{U}(g)(G(q))(\ell)(f(a))} & \text{distributivity of } \mathcal{R} \\
= & F(q)(\ell)(a) * \dfrac{\mathcal{U}(g)(G(q))(\ell)(f(a))}{\mathcal{U}(g)(G(q))(\ell)(f(a))} & \text{by definition of } \mathcal{U}(g) \\
= & F(q)(\ell)(a) &
\end{array}$$

Hence, also for this case, $F(q)(\ell)(a) = \mathcal{U}(\pi_1)(N(q))(\ell)(a)$. We conclude that $F = \mathcal{U}(\pi_1) \circ N$.

For proving the claim (3), it remains to check that $G = \mathcal{U}(\pi_2) \circ N$. Pick $q \in Q$, $\ell \in \mathcal{L}$ and $b \in B$. We now distinguish three cases.
Case I: $g(b) \notin f[A]$. Then, $\mathcal{U}(\pi_2)(N(q))(\ell)(b) = \sum_{a : f(a) = g(b)} N(q)(\ell)(a, b) = 0$ as the index set is empty. On the other hand,

$$\begin{array}{lll}
\multicolumn{2}{l}{\sum_{b' : g(b') = g(b)} G(q)(\ell)(b')} & \\
= & \sum_{b' \in g^{-1}(g(b))} G(q)(\ell)(b') & g(b') = g(b) \text{ iff } b' \in g^{-1}(g(b)) \\
= & \mathcal{U}(g)(G(q))(\ell)(g(b)) & \text{by definition of } \mathcal{U} \\
= & \mathcal{U}(f)(F(q))(\ell)(g(b)) & \text{since } \mathcal{U}(f) \circ F = \mathcal{U}(g) \circ G \\
= & \sum_{a' \in f^{-1}(g(b))} F(q)(\ell)(a) & \text{by definition of } \mathcal{U} \\
= & \sum_{a : f(a) = g(b)} F(q)(\ell)(a) & a' \in f^{-1}(g(b)) \text{ iff } f(a) = g(b) \\
= & 0 & \text{again because the index set is empty}
\end{array}$$

Since $\sum_{b':g(b')=g(b)} G(q)(\ell)(b') = 0$, it follows from the zero-sum property of $\mathcal{R}$ that $G(q)(\ell)(b') = 0$ for each $b' \in B$ such that $g(b') = g(b)$. In particular, $G(q)(\ell)(b) = 0$ and therefore $G(q)(\ell)(b) = \mathcal{U}(\pi_2)(N(q))(\ell)(b)$.

Case II: $g(b) = f(a)$ for some $a \in A$ and $\mathcal{U}(g)(G(q))(\ell)(f(a)) = 0$. Note that the latter equality is independent of the choice of a by equation (4). On the one hand, we have $\mathcal{U}(\pi_2)(N(q))(\ell)(b) = \sum_{a:f(a)=g(b)} N(q)(\ell)(a,b) = 0$, since by definition of N we have $N(q)(\ell)(a,b) = 0$ for each $a \in A$ such that $f(a) = g(b)$ in this case, as $\mathcal{U}(g)(G(q))(\ell)(f(a)) = 0$. On the other hand, $\sum_{b':g(b')=g(b)} G(q)(\ell)(b') = \mathcal{U}(g)(G(q))(\ell)(g(b)) = \mathcal{U}(g)(G(q))(\ell)(f(a)) = 0$ by assumption. Again, by the zero-sum property of $\mathcal{R}$, we obtain in particular that $G(q)(\ell)(b) = 0$. It follows that $G(q)(\ell)(b) = \mathcal{U}(\pi_2)(N(q))(\ell)(b)$.

Case III: $g(b) = f(a)$ for some $a \in A$ and $\mathcal{U}(g)(G(q))(\ell)(f(a)) \neq 0$. Then it holds that

$$\begin{array}{lll}
& \mathcal{U}(\pi_2)(N(q))(\ell)(b) & \\
= & \sum_{a \in \pi_2^{-1}(b)} N(q)(\ell)(a,b) & \text{by definition of } \mathcal{U}(\pi_2) \\
= & \sum_{a:f(a)=g(b)} N(q)(\ell)(a,b) & a \in \pi_2^{-1}(b) \text{ iff } f(a) = g(b) \\
= & \sum_{a:f(a)=g(b)} \dfrac{F(q)(\ell)(a) * G(q)(\ell)(b)}{\mathcal{U}(g)(G(q))(\ell)(f(a))} & \text{by definition of } N \\
= & \sum_{a \in f^{-1}(g(b))} \dfrac{F(q)(\ell)(a) * G(q)(\ell)(b)}{\mathcal{U}(g)(G(q))(\ell)(f(a))} & f(a) = g(b) \text{ iff } a \in f^{-1}(g(b)) \\
= & \dfrac{\sum_{a \in f^{-1}(g(b))} F(q)(\ell)(a)}{\mathcal{U}(f)(F(q))(\ell)(g(b))} * G(q)(\ell)(b) & \text{by equation (4) and distributivity of } \mathcal{R} \\
= & \dfrac{\mathcal{U}(f)(F(q))(\ell)(g(b))}{\mathcal{U}(f)(F(q))(\ell)(g(b))} * G(q)(\ell)(b) & \text{by definition of } \mathcal{U}(f) \\
= & G(q)(\ell)(b) &
\end{array}$$

Thus, also in this case $G(q)(\ell)(b) = \mathcal{U}(\pi_2)(N(q))(\ell)(b)$ and we conclude that $G = \mathcal{U}(\pi_2) \circ N$. This proves the claim.

Now, let W with span $w_1 : W \to A$ and $w_2 : W \to B$ be any weak pullback for f and g. See again Figure 4. Since W is a weak pullback of f and g, and π_1 and π_2 satisfy $f \circ \pi_1 = g \circ \pi_2$, there exists a mapping $m : P \to W$ such that $\pi_1 = w_1 \circ m$ and $\pi_2 = w_2 \circ m$. Suppose again that $F : Q \to \mathcal{U}(A)$ and $G : Q \to \mathcal{U}(B)$ is a span such that $\mathcal{U}(f) \circ F = \mathcal{U}(g) \circ G$. To prove the lemma we need to show that there exists $M : Q \to \mathcal{U}(W)$ such that $F = \mathcal{U}(w_1) \circ M$ and $G = \mathcal{U}(w_2) \circ M$. Put $M = \mathcal{U}(m) \circ N$. Clearly, $\mathcal{U}(\pi_1) = \mathcal{U}(w_1) \circ \mathcal{U}(m)$ and $\mathcal{U}(\pi_2) = \mathcal{U}(w_2) \circ \mathcal{U}(m)$. Therefore we have, by equation (3), $F = \mathcal{U}(\pi_1) \circ N = \mathcal{U}(w_1) \circ \mathcal{U}(m) \circ N = \mathcal{U}(w_1) \circ M$, and similarly $G = \mathcal{U}(w_2) \circ M$, as was to be shown. In [Bonchi et al.(2011), Section 2.2], in the setting of weighted automata, a counter example is given for the general case of Lemma 8. The construction there involves the semiring $\mathbb{Z}$ and, specifically the fact that $1 + (-1) = 0$. Thus the construction of [Bonchi et al.(2011)] exploits in particular that the semiring involved does not have the zero-sum property.

The proof of Lemma 8, in line with [de Vink & Rutten(1999)], explictly constgructs the mediating morphism N. One can also appeal to a generalization of the so-called Row-Column Theorem, as coined by Moss in [Moss(1999)]. An outline of such a proof is sketched in [Klin(2009)]. For a version of the Row/Column Theorem dealing with infinite matrices, see [Sokolova(2005), Lemma 3.5.5]. In a setting with finite sums, [Gumm & Schröder(2001)] provides a characterization of the monoids $\mathcal{M}$ for which the functor $\mathcal{FS}(\cdot, \mathcal{M})^{\mathcal{L}}$ preserves weak pullbacks, viz. those monoids with have the zero-sum property and for which the Row/Column Theorem holds too.

With the Lemma 8 in place we are in a position to relate behavioural equivalence and coalgebraic bisimulation as induced by a *FuTS*. Note, as for Lemma 8, the proof relies on the semirings to have the zero-sum property.

Theorem 9 *Let $\mathcal{S} = (\, S,\, \langle\, \theta_i \,\rangle_{i=1}^{n}\,)$ be a FuTS over the label sets $\mathcal{L}_i$ and semirings $\mathcal{R}_i$, $i = 1 \ldots n$, with multiplicative inverse and satisfying the zero-sum property. For $\mathcal{V}$ as in Definition 4 it holds that $\approx_{\mathcal{V}}$ and $\sim_{\mathcal{V}}$ coincide.*

Proof We recall that the functor $\mathcal{V}$ on **Set** is defined by $\mathcal{V} = \prod_{i=1}^{n} \mathcal{U}_i$, where, for $i = 1 \ldots n$, $\mathcal{U}_i = \mathcal{FS}(\,\cdot\,, \mathcal{R}_i\,)^{\mathcal{L}_i}$. By Lemma 8, each factor $\mathcal{U}_i$, $i = 1 \ldots n$, preserves weak pullbacks.

Let $f : A \to C$ and $g : B \to C$ be a cospan and let W with $w_1 : W \to A$ and $w_2 : W \to B$ be a weak pullback for f and g. Suppose $F : Q \to \mathcal{V}(A)$ and $G : Q \to \mathcal{V}(B)$ is a span such that $\mathcal{V}(f) \circ F = \mathcal{V}(g) \circ G$. See Figure 5.

By Lemma 8 we have, for all $i = 1 \ldots n$, that $\mathcal{U}_i(W)$ with $\mathcal{U}_i(w_1)$ and $\mathcal{U}_i(w_2)$ is a weak pullback of $\mathcal{U}_i(f)$ and $\mathcal{U}_i(g)$. Since, for the projections $\pi_i^A : \mathcal{V}(A) \to \mathcal{U}_i(A)$, $\pi_i^B : \mathcal{V}(B) \to \mathcal{U}_i(B)$ and $\pi_i^C : \mathcal{V}(C) \to \mathcal{U}_i(C)$, we have $\mathcal{U}_i(f) \circ \pi_i^A \circ F = \pi_i^C \circ \mathcal{V}(f) \circ F = \pi_i^C \circ \mathcal{V}(g) \circ G = \mathcal{U}_i(g) \circ \pi_i^B \circ G$, there exists a mapping $n_i : Q \to \mathcal{U}_i(W)$ such that $\pi_i^A \circ F = \mathcal{U}_i(w_1) \circ n_i$ and $\pi_i^B \circ G = \mathcal{U}_i(w_2) \circ n_i$ by the weak pullback property, for each index $i = 1 \ldots n$. Define $N : Q \to \mathcal{V}(W)$ by $N = n_1 \times \cdots \times n_n$. Then we have, for any $q \in Q$,

$$\begin{aligned}
(\mathcal{V}(w_1) \circ N)(q) &= \mathcal{V}(w_1)(\,\langle\, n_1(q), \ldots, n_n(q) \,\rangle\,) = \\
&\langle\, \mathcal{U}_1(w_1)(n_1(q)), \ldots, \mathcal{U}_n(w_1)(n_n(q)) \,\rangle = \langle \pi_1^A(F(q)), \ldots, \pi_n^A(F(q)) \rangle = F(q)
\end{aligned}$$

Thus $\mathcal{V}(w_1) \circ N = F$. Similarly, $\mathcal{V}(w_2) \circ N = G$. We conclude that $\mathcal{V}(W)$ with $\mathcal{V}(w_1)$ and $\mathcal{V}(w_2)$ is a weak pullback of $\mathcal{V}(f)$ and $\mathcal{V}(g)$ and that the functor $\mathcal{V}$ preserves weak pullbacks. Therefore, by Theorem 4, the result follows.

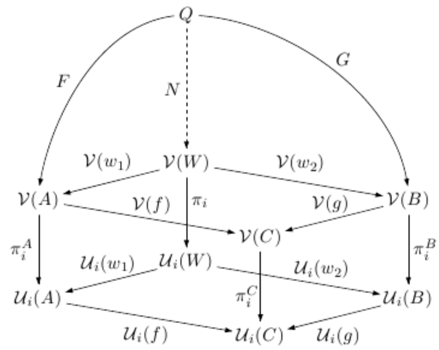

Figure 5: Functor $\mathcal{V}$ preserves weak pullbacks too, since each factor $\mathcal{U}_i$ does

In the sequel we will provide a *FuTS* semantics for three representative process languages, one with qualitative non-determinacy, one quantitative with stochastic-time non-determinacy, and a mixed one having both qualitative and quantitative stochastic-time non-determinacy. For these languages we will establish that their standard notions of strong equivalence as known in the literature coincide with the notion of strong bisimulation as induced by the *FuTS* semantics. The results of this section imply that the standard notions of strong equivalence on the one hand, and behavioural equivalence and coalgebraic bisimulation on the other hand, are all the same. The notion of bisimulation for *FuTS* plays an intermediary role, it bridges between the standard notion of concrete equivalence and the abstraction notions from coalgebra.

5 *FuTS*-semantics for a elementary process language

As a first example of a formal semantics based on *FuTS*s and of an illustration of the comparison with standard semantics using *LTS*s, we first consider a minimal process language common to many process algebras [Hoare(1985), Milner(1989), Baeten et al.(2009)] involving action prefix, non-deterministic choice and recursion via process variables. Then we consider an extension of the language with a parallel operator. The language comprises the basis for many of the SPCs proposed in the literature, and in particular those we discuss in Sections 6 and 7. In fact, the language is a sublanguage of IML (see Section 7).

Definition 5 *Let $\mathcal{A}$ be a set of actions, ranged over by a, and let $\mathcal{X}$ be the set of constants, or process variables, ranged over by X. The set $\mathcal{P}_{elm}$ of 'elementary' process terms is given by the grammar* $P \ ::= \ \textbf{\textit{nil}} \mid a.P \mid P + P \mid X$. •

We associate with each $X \in \mathcal{X}$ a unique process $P \in \mathcal{P}_{elm}$, notation $X := P$. It is required that each occurrence of a constant in the body P of the constant definition is in the scope of a prefix.

The semantics of elementary processes is given as a *FuTS* over the action set $\mathcal{A}$ and the semiring of booleans $\mathbb{B}$.

Definition 6 *The FuTS semantics of $\mathcal{P}_{elm}$ is given by the simple FuTS $\mathcal{S}_{elm} = (\mathcal{P}_{elm}, \rightarrowtail_{elm})$ where the transition relation $\rightarrowtail_{elm} \subseteq \mathcal{P}_{elm} \times \mathcal{A} \times \mathcal{FS}(\mathcal{P}_{elm}, \mathbb{B})$ is the least relation satisfying the rules of Figure 6.* •

$$\text{(NIL)}\ \frac{}{\textbf{nil} \overset{a}{\rightarrowtail}_{elm} []_{\mathbb{B}}} \qquad \text{(PREF1)}\ \frac{}{a.P \overset{a}{\rightarrowtail}_{elm} [P \mapsto \texttt{true}]} \qquad \text{(PREF2)}\ \frac{a \neq b}{a.P \overset{b}{\rightarrowtail}_{elm} []_{\mathbb{B}}}$$

$$\text{(CHO)}\ \frac{P \overset{a}{\rightarrowtail}_{elm} \mathscr{P} \quad Q \overset{a}{\rightarrowtail}_{elm} \mathscr{Q}}{P + Q \overset{a}{\rightarrowtail}_{elm} \mathscr{P} + \mathscr{Q}} \qquad \text{(CNS)}\ \frac{P \overset{a}{\rightarrowtail}_{elm} \mathscr{P} \quad X := P}{X \overset{a}{\rightarrowtail}_{elm} \mathscr{P}}$$

Figure 6: *FuTS* semantics for elementary processes.

The **nil** process does not display any activity. Therefore, for every action $a \in \mathcal{A}$, the function that records the possible continuations of **nil** has an empty support, i.e. $\textbf{nil} \overset{a}{\rightarrowtail}_{elm} []_{\mathbb{B}}$, which is exactly rule (NIL). Recall, `false` is the 0-element of the semiring $\mathbb{B}$ and $[]_{\mathbb{B}} : \mathcal{P}_{elm} \to \mathbb{B}$ is defined by $[]_{\mathbb{B}}(P) = \texttt{false}$ for every $P \in \mathcal{P}_{elm}$. An action-prefix process $a.P$ executes a and continues to behave as P. This is captured by rule (PREF1). For any other action b, there is no continuation of $a.P$, see rule (PREF2). With respect to the action b the processes $a.P$ and **nil** behave the same.

In dealing with non-deterministic choice we take advantage of the additive structure of the semiring, that extends pointwise to functions. If $P_1, \ldots, P_n$ are all possible continuations for the process P after executing a, and $Q_1, \ldots, Q_m$ are all possible continuations for the process Q after executing a, then after doing a the process $P + Q$ has the possibilities $P_1, \ldots, P_n$ as well as $Q_1, \ldots, Q_m$. In case $\mathscr{P} = [P_i \mapsto \texttt{true}]_{i=1}^n$ and $\mathscr{Q} = [Q_j \mapsto \texttt{true}]_{j=1}^m$, this is represented in rule (CHO) by the function $\mathscr{P} + \mathscr{Q} = [P_i \mapsto \texttt{true}]_{i=1}^n + [Q_j \mapsto \texttt{true}]_{j=1}^m = [P_1 \mapsto \texttt{true}, \ldots, P_n \mapsto \texttt{true}, Q_1 \mapsto \texttt{true}, \ldots, Q_m \mapsto \texttt{true}]$. The rule (CNS) for the constant simply copies the transition for the 'body' P of the constant X if we have $X := P$. Note, that such is well-defined as P is required to be guarded.

It can be straightforwardly shown by guarded induction that the *FuTS* $\mathcal{S}_{elm}$ is total and deterministic: Clearly, for each $a \in \mathcal{A}$ there exists a unique $\mathscr{P} \in \mathcal{FS}(\mathcal{P}_{elm}, \mathbb{B})$ with $\textbf{nil} \overset{a}{\rightarrowtail}_{elm} \mathscr{P}$, viz. $[]_{\mathbb{B}}$. Also, for each $b \in \mathcal{A}$, either $b = a$ or $b \neq a$, there exists a unique $\mathscr{P} \in \mathcal{FS}(\mathcal{P}_{elm}, \mathbb{B})$ with $a.P \overset{b}{\rightarrowtail}_{elm} \mathscr{P}$. In case $a = b$ we have $\mathscr{P} = [P \mapsto \texttt{true}]$, in case $a \neq b$ we have $\mathscr{P} = []_{\mathbb{B}}$. Assuming there exist, for given $a \in \mathcal{A}$, unique $\mathscr{P}$ and $\mathscr{Q}$ such that $P \overset{a}{\rightarrowtail}_{elm} \mathscr{P}$ and $Q \overset{a}{\rightarrowtail}_{elm} \mathscr{Q}$, respectively, it follows, since only rule (CHO) applies, that there exists a unique $\mathscr{P} \in \mathcal{FS}(\mathcal{P}_{elm}, \mathbb{B})$, namely $\mathscr{P} + \mathscr{Q}$ that aggregates the results for P, given by $\mathscr{P}$, and the results for Q, given by $\mathscr{Q}$. Finally, for $X := P$, by the induction hypothesis there exists a unique $\mathscr{P}$ such that $P \overset{a}{\rightarrowtail}_{elm} \mathscr{P}$, for $a \in \mathcal{A}$. Then, by virtue of rule (CNS), $\mathscr{P}$ is also unique such that $X \overset{a}{\rightarrowtail}_{elm} \mathscr{P}$.

In the following we will use $\theta_{elm} : \mathcal{P}_{elm} \to \mathcal{A} \to \mathcal{FS}(\mathcal{P}_{elm}, \mathbb{B})$ for the function corresponding to the relation $\rightarrowtail_{elm}$. We have, $P \overset{a}{\rightarrowtail}_{elm} \mathscr{P}$ iff $\theta_{elm}(P)(a) = \mathscr{P}$.

In Figure 7 we provide the SOS for elementary processes in the LTS-based approach defining the transition relation $\rightarrow_{elm}$. We have $\rightarrow_{elm} \subseteq \mathcal{P}_{elm} \times \mathcal{A} \times \mathcal{P}_{elm}$. We discuss the various differences of the *FuTS* and *LTS* semantics. In the standard semantics there is no rule for the **nil** process, i.e. there is no transition for **nil**. The *FuTS* semantics provides $\mathbf{nil} \overset{a}{\rightarrowtail}_{elm} []_{\mathbb{B}}$ for every action a. However, the latter expresses $\theta_{elm}(\mathbf{nil})(a)(P') = 0$ for every $a \in \mathcal{A}$ and $P' \in \mathcal{P}_{elm}$, or, in standard terminology, **nil** has no transition. The standard approach provides only one rule for the prefix construct, rule (PREF). The *FuTS* approach has two rules for this, rule (PREF1) and (PREF2), as it also explicitly expresses by way of rule (PREF2) that there is no standard transition labeled with another label than a for the process $a.P$.

The choice rules (CHOa) and (CHOb) for the *LTS* semantics similarly differ from the choice rule (CHO) for the *FuTS* semantics. In the standard approach the non-determinism is resolved by choosing the rule. The *FuTS* semantics, developed with stochastic process languages in mind where multiplicities matter (see Sections 6 and 7), combines the two branches into one function. The treatment of process variables is the same for the two approaches.

$$\text{(PREF)}\ \frac{}{a.P \overset{a}{\rightarrow}_{elm} P} \qquad \text{(CNS)}\ \frac{P \overset{a}{\rightarrow}_{elm} P' \quad X := P}{X \overset{a}{\rightarrow}_{elm} P'}$$

$$\text{(CHOa)}\ \frac{P \overset{a}{\rightarrow}_{elm} P'}{P + Q \overset{a}{\rightarrow}_{elm} P'} \qquad \text{(CHOb)}\ \frac{Q \overset{a}{\rightarrow}_{elm} Q'}{P + Q \overset{a}{\rightarrow}_{elm} Q'}$$

Figure 7: Standard SOS for elementary processes.

As we will show, see Theorem 11, for $\mathcal{S}_{elm} = (\mathcal{P}_{elm}, \rightarrowtail_{elm})$ like for other *FuTS*s to follow, the standard notion of strong bisimulation [Park(1981), Milner(1980)] identifies the same processes as $\mathcal{S}_{elm}$-bisimulation as obtained from Definition 2 and denoted by $\simeq_{\mathcal{S}_{elm}}$. We first observe the following lemma.

Lemma 10 *Let $P \in \mathcal{P}_{elm}$ and $a \in \mathcal{A}$. Suppose $P \overset{a}{\rightarrowtail}_{elm} \mathscr{P}$. Then $\mathscr{P}(R) \iff P \overset{a}{\rightarrow} R$, for all $R \in \mathcal{P}_{elm}$.*
Proof *Guarded induction on P: Clear for **nil**, as $\mathscr{P}(R) = []_{\mathbb{B}}(R) = \mathtt{false}$ and $\mathbf{nil} \overset{a}{\rightarrow}_{elm} R$ for no $R \in \mathcal{P}_{elm}$. For $b.P$, we have $\mathscr{P}(R)$ iff $b = a$ and $[P \mapsto \mathtt{true}](R)$ iff $b = a$ and $P = R$, and also $b.P \overset{a}{\rightarrow}_{elm} R$ iff $b = a$ and $P = R$. For $P + Q$, it holds that $\mathscr{P} = \mathscr{P}' + \mathscr{P}''$ where $P \overset{a}{\rightarrowtail}_{elm} \mathscr{P}'$ and $Q \overset{a}{\rightarrowtail}_{elm} \mathscr{P}''$. Thus $\mathscr{P}(R)$ iff $(\mathscr{P}' + \mathscr{P}'')(R)$ iff $\mathscr{P}'(R)$ or $\mathscr{P}''(R)$ iff, by induction hypothesis, $P \overset{a}{\rightarrow}_{elm} R$ or $Q \overset{a}{\rightarrow}_{elm} R$ iff $P + Q \overset{a}{\rightarrow}_{elm} R$. For X with $X := P$, we have $X \overset{a}{\rightarrowtail}_{elm} \mathscr{P}$ iff $P \overset{a}{\rightarrowtail}_{elm} \mathscr{P}$, and $X \overset{a}{\rightarrow}_{elm} R$ iff $P \overset{a}{\rightarrow}_{elm} R$. Thus $\mathscr{P}(R)$ iff, by induction hypothesis, $P \overset{a}{\rightarrow}_{elm} R$ iff $X \overset{a}{\rightarrow}_{elm} R$.*

Now, following the usual terminology, a relation $R \subseteq \mathcal{P}_{elm} \times \mathcal{P}_{elm}$ is called a strong bisimulation on $\mathcal{P}_{elm}$ if $P_1 \overset{a}{\rightarrow}_{elm} P_1' \Rightarrow \exists P_2' \colon P_2 \overset{a}{\rightarrow}_{elm} P_2' \wedge R(P_1', P_2')$ and $P_2 \overset{a}{\rightarrow}_{elm} P_2' \Rightarrow \exists P_1' \colon P_1 \overset{a}{\rightarrow}_{elm}$

$P_1' \wedge R(P_1', P_2')$ whenever $R(P_1, P_2)$. Two elements $P_1, P_2 \in \mathcal{P}_{elm}$ are called strongly bisimilar if there exists a strong bisimulation R on $\mathcal{P}_{elm}$ relating P_1 and P_2, notation $P_1 \sim_{elm} P_2$. It is well-known that we can require R to be an equivalence relation, a so-called strong bisimulation equivalence, as the equivalence closure of R is a strong bisimulation if R is.

In order to relate strong bisimilarity and *FuTS*-bisimilarity we observe the following: If R is a strong bisimulation equivalence, with equivalence classes $[P]_R$ for $P \in \mathcal{P}_{elm}$, then it holds for $P_1, P_2 \in X$ with $R(P_1, P_2)$ that

$$\exists P' \in [P]_R \colon P_1 \xrightarrow{a}_{elm} P' \iff \exists P'' \in [P]_R \colon P_2 \xrightarrow{a}_{elm} P'' \tag{5}$$

For, if $P_1 \xrightarrow{a}_{elm} P'$ and $R(P', P)$, then, since R is a bisimulation, for some suitable P'' we have $P_2 \xrightarrow{a}_{elm} P''$, $R(P', P'')$ and $R(P', P)$, but then also $P_2 \xrightarrow{a}_{elm} P''$ and $R(P'', P)$, since R is an equivalence. We use this in the proof that strong bisimilarity $\sim_{elm}$ and *FuTS*-bisimilarity $\simeq_{\mathcal{S}_{elm}}$ for $\mathcal{S}_{elm}$ coincide.

Theorem 11 *For any two processes $P_1, P_2 \in \mathcal{P}_{elm}$, it holds that $P_1 \sim_{elm} P_2$ iff $P_1 \simeq_{\mathcal{S}_{elm}} P_2$.*
Proof *Pick $P_1, P_2 \in \mathcal{P}_{elm}$. To verify that $P_1 \sim_{elm} P_2$ implies $P_1 \simeq_{\mathcal{S}_{elm}} P_2$ we assume there exists an equivalence relation R that is a strong bisimulation such that $R(P_1, P_2)$. Then, for arbitrary $a \in \mathcal{A}$, we have*

$$\begin{array}{lll}
\sum_{Q' \in [Q]_R} \theta_{elm}(P_1)(a)(Q') & & \\
\iff & \exists Q' \in [Q]_R \colon \theta_{elm}(P_1)(a)(Q') & \\
\iff & \exists Q' \in [Q]_R \colon P_1 \xrightarrow{a}_{elm} Q' & \textit{by Lemma 10} \\
\iff & \exists Q'' \in [Q]_R \colon P_2 \xrightarrow{a}_{elm} Q'' & \textit{by equation (5)} \\
\iff & \exists Q' \in [Q]_R \colon P_2 \xrightarrow{a}_{elm} Q' & \alpha\textit{-conversion} \\
\iff & \exists Q' \in [Q]_R \colon \theta_{elm}(P_2)(a)(Q') & \textit{by Lemma 10} \\
\iff & \sum_{Q' \in [Q]_R} \theta_{elm}(P_2)(a)(Q') &
\end{array}$$

for all $Q \in \mathcal{P}_{elm}$, i.e. for all equivalence classes of R. Hence, R is also an $\mathcal{S}_{elm}$-bisimulation and, since $R(P_1, P_2)$, we conclude $P_1 \simeq_{\mathcal{S}_{elm}} P_2$.

To show the reverse, that $P_1 \simeq_{\mathcal{S}_{elm}} P_2$ implies $P_1 \sim_{elm} P_2$, let R be an $\mathcal{S}_{elm}$-bisimulation with $R(P_1, P_2)$. We have

$$\begin{array}{lll}
P_1 \xrightarrow{a}_{elm} Q' & & \\
\implies & \theta_{elm}(P_1)(a)(Q') & \textit{by Lemma 10} \\
\implies & \sum_{Q \in [Q']_R} \theta_{elm}(P_1)(a)(Q) & \textit{property } \mathbb{B} \\
\implies & \sum_{Q \in [Q']_R} \theta_{elm}(P_2)(a)(Q) & R \textit{ is an } \mathcal{S}_{elm}\textit{-bisimulation} \\
\implies & \exists Q \in [Q']_R \colon \theta_{elm}(P_2)(a)(Q) & \textit{property } \mathbb{B} \\
\implies & \exists Q'' \in [Q']_R \colon \theta_{elm}(P_2)(a)(Q'') & \alpha\textit{-conversion} \\
\implies & \exists Q'' \colon P_2 \xrightarrow{a}_{elm} Q'' \wedge R(Q', Q'') & \textit{by Lemma 10}
\end{array}$$

Thus, the first transfer condition is met. By symmetry it follows that R is a strong bisimulation on $\mathcal{P}_{elm}$ and $R(P_1, P_2)$. Thus $P_1 \sim_{elm} P_2$.

Next we augment the collection of elementary processes with a(n extension of a) CSP-like parallel operator. Its treatment is slightly involved since a little extra semantic machinery for continuation functions to reflect the syntactic operator is needed. As this obscures a bit the general explanation, we kept initially the parallel construct out of Definition 5, as our main purpose of considering a qualitative process language here is to illustrate the overall approach with *FuTS* in the quantitative setting.

$$\text{(PAR1)}\ \frac{P \overset{a}{\rightarrowtail}_{elm} \mathscr{P} \quad Q \overset{a}{\rightarrowtail}_{elm} \mathscr{Q} \quad a \notin \mathcal{A}}{P \parallel_A Q \overset{a}{\rightarrowtail}_{elm} (\, \mathscr{P} \parallel_A \mathbf{D}_Q \,) + (\, \mathbf{D}_P \parallel_A \mathscr{Q} \,)}$$

$$\text{(PAR2)}\ \frac{P \overset{a}{\rightarrowtail}_{elm} \mathscr{P} \quad Q \overset{a}{\rightarrowtail}_{elm} \mathscr{Q} \quad a \in \mathcal{A}}{P \parallel_A Q \overset{a}{\rightarrowtail}_{elm} \mathscr{P} \parallel_A \mathscr{Q}}$$

Figure 8: *FuTS* semantics for the parallel operator $\parallel_A$

We extend $\mathcal{P}_{elm}$ to allow processes of the form $P \parallel_A Q$, for every subset A of the set of actions $\mathcal{A}$. The computational intuition is that in $P \parallel_A Q$ the processes P and Q interleave a-steps for each action $a \notin \mathcal{A}$ and that P and Q synchronise a-steps when the action $a \in \mathcal{A}$.

We redefine the process language $\mathcal{P}_{elm}$ to be given by the grammar

$$P \ ::= \ \mathbf{nil} \ \mid \ a.P \ \mid \ P{+}P \ \mid \ P \parallel_A P \ \mid \ X$$

for $a \in \mathcal{A}$, $A \subseteq \mathcal{A}$ and $X \subseteq \mathcal{X}$. To handle the parallel construct $P \parallel_A Q$ with *FuTS*s we extend the state-to-function relation $\rightarrowtail_{elm}$ by adding the operational rules (PAR1) and (PAR2) of Figure 8.

Rule (PAR1) treats the case where the two parallel operands do not synchronise. If the process $P \parallel_A Q$ executes an action $a \in \mathcal{A}$, this either stems from P or from Q. In the former case Q remains as is, in the latter case P. Following the description introduced in Section 2 for injective binary operation $\parallel_A$ on process terms, we here consider

$$\parallel_A : (\mathcal{P}_{elm} \to \mathcal{FS}(\mathcal{P}_{elm}, \mathbb{B})) \times (\mathcal{P}_{elm} \to \mathcal{FS}(\mathcal{P}_{elm}, \mathbb{B})) \to (\mathcal{P}_{elm} \to \mathcal{FS}(\mathcal{P}_{elm}, \mathbb{B}))$$

as the semantical counterpart, by putting $(\mathscr{P}_2 \parallel_A \mathscr{P}_2)(R) = \mathscr{P}_1(R_1) \wedge \mathscr{P}_2(R_2)$ if $R = R_1 \parallel_A R_2$. More specifically, for any subset $A \subseteq \mathcal{A}$,

$$\begin{aligned}
&(\,[\,P_1 \mapsto \texttt{true}, \ldots, P_n \mapsto \texttt{true}\,] \parallel_A [\,Q_1 \mapsto \texttt{true}, \ldots, Q_m \mapsto \texttt{true}\,])(R)\\
&= \ \textstyle\sum_{R_1,R_2:\, R_1 \parallel_A R_2 = R} [\,P_1 \mapsto \texttt{true}, \ldots, P_n \mapsto \texttt{true}\,](R_1) * [\,Q_1 \mapsto \texttt{true}, \ldots, Q_m \mapsto \texttt{true}\,](R_2)\\
&= \ \textstyle\bigvee_{R_1,R_2:\, R_1 \parallel_A R_2 = R} (\bigvee_{i=1}^{n} (R_1 = P_i)) \wedge (\bigvee_{j=1}^{m} (R_2 = Q_j))\\
&= \ \textstyle\bigvee_{i=1}^{n} \bigvee_{j=1}^{m} (R = P_i \parallel_A Q_j)
\end{aligned}$$

Note, if existent, a split-up $R = R_1 \parallel_A R_2$ is unique here, as we do not deal with congruence classes directly, cf. [Cardelli & Mardare(2010)]. For the carrier $\mathcal{P}_{elm}$ and the semiring $\mathbb{B}$ we have the Dirac functions $\mathbf{D}_Q : \mathcal{P}_{elm} \to \mathbb{B}$ and $\mathbf{D}_Q = [Q \mapsto \texttt{true}]$, for $Q \in \mathcal{P}_{elm}$. Thus, unfolding the various definitions, we get

$$\begin{aligned}
&((\, \mathscr{P} \parallel_A \mathbf{D}_Q \,) + (\, \mathbf{D}_P \parallel_A \mathscr{Q} \,))(R)\\
\iff\ & \text{by definition of } + \text{ on } \mathcal{FS}(\mathcal{P}_{elm}, \mathbb{B})\\
& (\, \mathscr{P} \parallel_A \mathbf{D}_Q \,)(R) \vee (\, \mathbf{D}_P \parallel_A \mathscr{Q} \,)(R)\\
\iff\ & \text{by definition of } \parallel_A \text{ on } \mathcal{FS}(\mathcal{P}_{elm}, \mathbb{B})\\
& (\,(R = R_1 \parallel_A R_2) \wedge \mathscr{P}(R_1) \wedge \mathbf{D}_Q(R_2)\,) \vee (\,(R = R_1 \parallel_A R_2) \wedge \mathbf{D}_P(R_1) \wedge \mathscr{Q}(R_2)\,)\\
\iff\ & (\,(R = R_1 \parallel_A Q) \wedge \mathscr{P}(R_1)\,) \vee (\,(R = P \parallel_A R_2) \wedge \mathscr{Q}(R_2)\,)
\end{aligned}$$

In other words, if $P \overset{a}{\rightarrowtail}_{elm} \mathscr{P}$ and $Q \overset{a}{\rightarrowtail}_{elm} \mathscr{Q}$ then $((\, \mathscr{P} \parallel_A \mathbf{D}_Q \,) + (\, \mathbf{D}_P \parallel_A \mathscr{Q} \,))(R) = \texttt{true}$ iff $R = R_1 \parallel_A Q$ and $\mathscr{P}(R_1) = \texttt{true}$ or $R = P \parallel_A R_2$ and $\mathscr{Q}(R_2) = \texttt{true}$.

Rule (PAR1) deals with interleaving for actions $a \notin \mathcal{A}$ only. Synchronization of P and of Q for $P \parallel_A Q$ on actions $a \in A$ is incorporated in rule (PAR2). As both process P and process Q need to progress, with respect to the action $a \in A$, the continuation $\mathscr{P}$ for P and the continuation $\mathscr{Q}$ for Q can be combined directly. For the extended state-to-function transition relation totality and determinacy is straightforward to check.

Example Consider the process $P = a.P_1 + b.P_2 + c.P_3 + d.P_4$ and $Q = a.Q_1 + b.Q_2 + b.Q_3 + c.Q_4$. Let $A = \{a, b\}$. Then we have

$$
\begin{array}{ll}
P \xrightarrowtail{a}_{elm} [P_1 \mapsto \mathtt{true}] & Q \xrightarrowtail{a}_{elm} [Q_1 \mapsto \mathtt{true}] \\
P \xrightarrowtail{b}_{elm} [P_2 \mapsto \mathtt{true}] & Q \xrightarrowtail{b}_{elm} [Q_2 \mapsto \mathtt{true},\, Q_3 \mapsto \mathtt{true}] \\
P \xrightarrowtail{c}_{elm} [P_3 \mapsto \mathtt{true}] & Q \xrightarrowtail{c}_{elm} [Q_4 \mapsto \mathtt{true}] \\
P \xrightarrowtail{d}_{elm} [P_4 \mapsto \mathtt{true}] & Q \xrightarrowtail{d}_{elm} []_{\mathbb{B}}
\end{array}
$$

Therefore, by rule (PAR2) for actions a and b and by rule (PAR1) for actions c and d, we have

$$
\begin{array}{l}
P \parallel_A Q \xrightarrowtail{a}_{elm} [\, P_1 \parallel_A Q_1 \mapsto \mathtt{true}\,] \\
P \parallel_A Q \xrightarrowtail{b}_{elm} [\, P_2 \parallel_A Q_2 \mapsto \mathtt{true},\, P_2 \parallel_A Q_3 \mapsto \mathtt{true}\,] \\
P \parallel_A Q \xrightarrowtail{c}_{elm} [\, P_3 \parallel_A Q \mapsto \mathtt{true},\, P \parallel_A Q_4 \mapsto \mathtt{true}\,] \\
P \parallel_A Q \xrightarrowtail{d}_{elm} [\, P_4 \parallel_A Q \mapsto \mathtt{true}\,]
\end{array}
$$

The standard SOS rules for the operator $\parallel_A$, for $A \subseteq \mathcal{A}$ are given in Figure 9: the two rules (PAR1a) and (PAR1b) for the interleaving case, the rule (PAR2) for the synchronising case. For the extended language of elementary processes Lemma 10 holds as well. For the corresponding extension of the proof only one other case for the induction step needs to be considered (split up in a subcase for an action $a \notin A$ and one for $a \in A$):

For the case $P_1 \parallel_A P_2$ where $P_1 \parallel_A P_2 \xrightarrowtail{a}_{elm} \mathscr{P}$ for $a \notin A$, suppose $P_1 \xrightarrowtail{a}_{elm} \mathscr{P}_1$ and $P_2 \xrightarrowtail{a}_{elm} \mathscr{P}_2$. We have, for $R \in \mathcal{P}_{elm}$,

$$
\begin{array}{lll}
\mathscr{P}(R) & \iff & \text{rule (PAR1) for } \rightarrowtail_{elm} \\
& & (\, \mathscr{P}_1 \parallel_A \mathbf{D}_{P_2}\,)(R) \vee (\, \mathbf{D}_{P_1} \parallel_A \mathscr{P}_2\,)(R) \\
& \iff & \text{definition } \parallel_A \text{ and } \mathbf{D}_{P_1}, \mathbf{D}_{P_2} \\
& & (\, \exists R_1 \colon R = R_1 \parallel_A P_2 \,\wedge\, \mathscr{P}_1(R_1)\,) \vee (\, \exists R_2 \colon R = P_1 \parallel_A R_2 \,\wedge\, \mathscr{P}_2(R_2)\,) \\
& \iff & \text{by induction hypothesis} \\
& & (\, \exists R_1 \colon R = R_1 \parallel_A P_2 \,\wedge\, P_1 \xrightarrow{a}_{elm} R_1\,) \vee (\, \exists R_2 \colon R = P_1 \parallel_A R_2 \,\wedge\, P_2 \xrightarrow{a}_{elm} R_2\,) \\
& \iff & \text{both (PAR1) rules for } \rightarrow_{elm} \\
& & (\, \exists R_1 \colon R = R_1 \parallel_A P_2 \,\wedge\, P_1 \parallel_A P_2 \xrightarrow{a}_{elm} R_1 \parallel_A P_2\,) \vee \\
& & \qquad\qquad (\, \exists R_2 \colon R = P_1 \parallel_A R_2 \,\wedge\, P_1 \parallel_A P_2 \xrightarrow{a}_{elm} P_1 \parallel_A R_2\,) \\
& \iff & P_1 \parallel_A P_2 \xrightarrow{a}_{elm} R
\end{array}
$$

For the case $P_1 \parallel_A P_2$ where $P_1 \parallel_A P_2 \xrightarrowtail{a}_{elm} \mathscr{P}$ for $a \in A$, suppose $P_1 \xrightarrowtail{a}_{elm} \mathscr{P}_1$ and $P_2 \xrightarrowtail{a}_{elm} \mathscr{P}_2$. We have, for $R \in \mathcal{P}_{elm}$,

$$\text{(PAR1a)}\ \frac{P \xrightarrow{a}_{elm} P' \quad a \notin \mathcal{A}}{P \parallel_A Q \xrightarrow{a}_{elm} P' \parallel_A Q} \qquad \text{(PAR1b)}\ \frac{Q \xrightarrow{a}_{elm} Q' \quad a \notin \mathcal{A}}{P \parallel_A Q \xrightarrow{a}_{elm} P \parallel_A Q'}$$

$$\text{(PAR2)}\ \frac{P \xrightarrow{a}_{elm} P \quad Q \xrightarrow{a}_{elm} Q \quad a \in \mathcal{A}}{P \parallel_A Q \overset{a}{\rightarrowtail}_{elm} P' \parallel_A Q'}$$

Figure 9: Standard semantics for the parallel operator $\parallel_A$

$$\begin{array}{cl}
\mathscr{P}(R) & \\
\iff & \text{rule (PAR2) for } \rightarrowtail_{elm} \\
& (\ \mathscr{P}_1 \parallel_A \mathscr{P}_2\ )(R) \\
\iff & \text{definition } \parallel_A \text{ on } \mathcal{FS}(\mathcal{P}_{elm}, \mathbb{B}\ ) \\
& \exists R_1, R_2 \colon R = R_1 \parallel_A R_2 \ \wedge\ \mathscr{P}_1(R_1) \ \wedge\ \mathscr{P}_2(R_2) \\
\iff & \text{by induction hypothesis} \\
& \exists R_1, R_2 \colon R = R_1 \parallel_A R_2 \ \wedge\ P_1 \xrightarrow{a}_{elm} R_1 \ \wedge\ P_2 \xrightarrow{a}_{elm} R_2 \\
\iff & \text{rule (PAR2) for } \rightarrow_{elm} \\
& \exists R_1 \colon R = R_1 \parallel_A R_2 \ \wedge\ P_1 \parallel_A P_2 \xrightarrow{a}_{elm} R_1 \parallel_A R_2 \\
\iff & P_1 \parallel_A P_2 \xrightarrow{a}_{elm} R
\end{array}$$

With the extension of Lemma 10 in place it follows that Theorem 11 is also valid for the extended set of processes.

Restriction and relabelling operators can be handled straightforwardly in the *FuTS* framework. A treatment of a CCS-style parallel operator with *FuTS* proceeds along the same lines, in particular when the set of actions $\mathcal{A}$ is assumed to be finite. For a countably infinite set $\mathcal{A}$, the complication arises that potentially unbounded sums need to be considered, as for every matching pair of actions a and $\bar{a}$ such that $P_1 \overset{a}{\rightarrowtail} \mathscr{P}_1$ and $P_2 \overset{\bar{a}}{\rightarrowtail} \mathscr{P}_2$ the combined continuation of $\mathscr{P}_1$ and $\mathscr{P}_2$ needs to be taken into account. It can be proven, but we do not so here, that the summations in the particular synchronization rule for the parallel operator are well-defined, using the fact that the processes involved admit finitely many transition only to a continuation different from $[]_{\mathbb{B}}$.

6 *FuTS* Semantics of *PEPA*

Next we will consider a significant part of the process algebra *PEPA* [Hillston(1996)], including the parallel operator implementing the scheme of so-called minimal apparent rates, and provide a *FuTS* semantics for it. We point out that there is no technical difficulty in extending the *FuTS* approach to the full language; we do not do so here since its treatment does not add any conceptual benefit to the present paper. We will show that *PEPA*'s notion of equivalence $\sim_{pepa}$, called strong equivalence in [Hillston(1996)], fits with the bisimilarity $\simeq_{\mathcal{S}_{pepa}}$ arising from the *FuTS* semantics.

Definition 7 *The set $\mathcal{P}_{pepa}$ of PEPA processes is given by the grammar below:*

$$P \ ::= \ \textbf{\textit{nil}} \mid (a, \lambda).P \mid P + P \mid P \underset{A}{\bowtie} P \mid X$$

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 $\mathcal{X}$. •

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

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

$$\text{(PAR1)}\ \frac{P \overset{\delta_a}{\rightarrowtail}_{pepa} \mathscr{P} \quad Q \overset{\delta_a}{\rightarrowtail}_{pepa} \mathscr{Q} \quad a \notin A}{P \underset{A}{\bowtie} Q \overset{\delta_a}{\rightarrowtail}_{pepa} (\mathscr{P} \underset{A}{\bowtie} \mathbf{D}_Q) + (\mathbf{D}_P \underset{A}{\bowtie} \mathscr{Q})} \qquad \text{(PAR2)}\ \frac{P \overset{\delta_a}{\rightarrowtail}_{pepa} \mathscr{P} \quad Q \overset{\delta_a}{\rightarrowtail}_{pepa} \mathscr{Q} \quad a \in A}{P \underset{A}{\bowtie} Q \overset{\delta_a}{\rightarrowtail}_{pepa} \mathrm{arf}(\mathscr{P},\mathscr{Q}) \cdot (\mathscr{P} \underset{A}{\bowtie} \mathscr{Q})}$$

Figure 10: *FuTS* semantics for *PEPA*.

For $X \in \mathcal{X}$, the notation $X := P$ indicates that the process P is associated with the constant X. It is required that each occurrence of a process constant in the body P of the definition $X := P$ is guarded by a prefix.

PEPA, like many other SPCs, e.g. [Hermanns et al.(1998), Bernardo & Gorrieri(1998)], 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 a random variable with an exponential distribution of rate λ. The CSP-like parallel composition $P \underset{A}{\bowtie} Q$ of a process P and a process Q for a set of actions $A \subseteq \mathcal{A}$ allows for the independent, asynchronous execution of actions of P or Q not occurring in the subset A, on the one hand, and requires the simultaneous, synchronised 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 in Figure 10, 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 [Hillston(1996)]. 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. The apparent rate $r_a(P)$ can easily be defined recursively on the structure of P (see [Hillston(1996), Definition 3.3.1] for details); accordingly, in the sequel we will refer to $r_a(P)$ as the 'syntactic' apparent rate. When considering the parallel composition $P \underset{A}{\bowtie} Q$, 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 (one cannot take the slowest process per sample, because such an operation cannot be expressed as an exponential distribution in general). Thus $r_a(P \underset{A}{\bowtie} Q)$ 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 $\mathit{arf}(P, Q) = 0$. Organising the product $r_1 \cdot r_2 \cdot \mathit{arf}(P, Q)$ differently as $r_1/r_a(P) \cdot r_2/r_a(Q) \cdot \min\{\, r_a(P),\ r_a(Q)\,\}$ we see that for $P \underset{A}{\bowtie} Q$ the minimum of the apparent rates $\min\{\, r_a(P),\ r_a(Q)\,\}$ 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.

The *FuTS* semantics of *PEPA* has been proposed originally in [De Nicola et al.(2011)]. The definition of the transition relation is recalled in Figure 10. The set of labels involved is $\Delta_{\mathcal{A}}$ defined by $\Delta_{\mathcal{A}} = \{\ \delta_a \mid a \in \mathcal{A}\,\}$. In the context of *FuTS* semantics considered in this paper, we conventionally use the special symbol δ for denoting a random *delay* with a negative exponential distribution. The symbol δ_a denotes the duration of the execution of the action a (assuming such a duration be random, exponentially distributed). The underlying semiring for the simple *FuTS* for *PEPA* is the semiring $\mathbb{R}_{\geqslant 0}$ of non-negative reals.

Definition 8 *The FuTS $\mathcal{S}_{pepa} = (\mathcal{P}_{pepa}, \rightarrowtail_{pepa})$ over $\Delta_{\mathcal{A}}$ and $\mathbb{R}_{\geqslant 0}$ has its transition relation given by the rules of Figure 10.* •

We discuss the rules of Figure 10. The *FuTS* semantics provides **nil** $\overset{\delta_a}{\rightarrowtail}_{pepa} []_{\mathbb{R}_{\geqslant 0}}$, for every action a, with $[]_{\mathbb{R}_{\geqslant 0}}$ the 0-function of $\mathbb{R}_{\geqslant 0}$. Therefore we have $\theta_{pepa}(\mathbf{nil})(\delta_a)(P') = 0$ for every $a \in \mathcal{A}$ 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 state-to-function 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). The treatment of constants is as usual.

Note the semantic sum of functions $\mathscr{P} + \mathscr{Q}$ replacing the syntactic sum in $P + Q$ and the semantic product $\mathscr{P} \underset{A}{\bowtie} \mathscr{Q}$ used in rules (PAR1) and (PAR2) and, we recall, is defined as follows, for all $R \in \mathcal{P}_{pepa}$:

$$(\mathscr{P} \underset{A}{\bowtie} \mathscr{Q})(R) = \begin{cases} \mathscr{P}(R_1) \cdot \mathscr{Q}(R_2) & \text{if } R = R_1 \underset{A}{\bowtie} R_2 \text{ for some } R_1, R_2 \in \mathcal{P}_{pepa} \\ 0 & \text{otherwise} \end{cases}$$

Note that the syntactic construct $\underset{A}{\bowtie}$ of *PEPA* is trivially injective. Regarding the parallel operator $\underset{A}{\bowtie}$, with respect to some cooperation set $A \subseteq \mathcal{A}$ there are again two rules. Now the distinction is between interleaving and synchronisation. 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 $P \underset{A}{\bowtie} Q$ makes progress. For example, the effect of the pattern $\mathscr{P} \underset{A}{\bowtie} \mathbf{D}_Q$ is that the value $\mathscr{P}(P') \cdot 1$ is assigned to a process $P' \underset{A}{\bowtie} Q$, the value $\mathscr{P}(P') \cdot 0 = 0$ to a process $P' \underset{A}{\bowtie} Q'$ for all $Q' \neq Q$, and the value 0 for a process not of the form $P' \underset{A}{\bowtie} Q'$. 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 $P \overset{\delta_a}{\rightarrowtail}_{pepa} \mathscr{P}$ and $Q \overset{\delta_a}{\rightarrowtail}_{pepa} \mathscr{Q}$, the 'semantic' scaling factor $\mathit{arf}(\mathscr{P}, \mathscr{Q})$ is applied to $\mathscr{P} \underset{A}{\bowtie} \mathscr{Q}$. This scaling factor defined for functions in $\mathcal{FS}(\mathcal{P}_{pepa}, \mathbb{R}_{\geqslant 0})$, is, very much similar to its 'syntactic' counterpart, given by

$$\mathrm{arf}(\mathscr{P}, \mathscr{Q}) = \frac{\min\{\oplus\mathscr{P}, \oplus\mathscr{Q}\}}{\oplus\mathscr{P} \cdot \oplus\mathscr{Q}}$$

provided $\oplus\mathscr{P}, \oplus\mathscr{Q} > 0$, and $\mathit{arf}(\mathscr{P}, \mathscr{Q}) = 0$ otherwise. For a process $R = R_1 \underset{A}{\bowtie} R_2$ we obtain the value $\mathit{arf}(\mathscr{P}, \mathscr{Q}) \cdot (\mathscr{P} \underset{A}{\bowtie} \mathscr{Q})(R_1 \underset{A}{\bowtie} R_2) = \mathit{arf}(\mathscr{P}, \mathscr{Q}) \cdot \mathscr{P}(R_1) \cdot \mathscr{Q}(R_2)$.

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

Lemma 12 *Let $P \in \mathcal{P}_{pepa}$ and $a \in \mathcal{A}$. Suppose $P \overset{\delta_a}{\rightarrowtail}_{pepa} \mathscr{P}$. Then $r_a(P) = \oplus\mathscr{P}$.* □

The proof of the lemma is straightforward (relying on the obvious definition of $r_a(P)$, omitted above, which can be found in [Hillston(1996)]). It is also easy to prove, by guarded induction, that the *FuTS* $\mathcal{S}_{pepa}$ given by Definition 8 is total and deterministic. So, it is justified to write $\mathcal{S}_{pepa} = (\mathcal{P}_{pepa}, \theta_{pepa})$. We use $\simeq_{\mathcal{S}_{pepa}}$ to denote the bisimilarity induced by $\mathcal{S}_{pepa}$.

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

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

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

Figure 11: Standard semantics for *PEPA*.

Lemma 13 *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 that $P_1 \overset{\delta_a}{\rightarrowtail}_{pepa} [\, P \mapsto \lambda \,]$ by rule (RAPF1), but $P_2 \overset{\delta_a}{\rightarrowtail}_{pepa} [\, P \mapsto 2\lambda \,]$ 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 $P_1 \not\simeq_{\mathcal{S}_{pepa}} P_2$. 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* [Hillston(1996), Hillston(2005)] is given in Figure 11. The transition relation $\rightarrow_{pepa} \subseteq \mathcal{P}_{pepa} \times (\mathcal{A} \times \mathbb{R}_{>0}) \times \mathcal{P}_{pepa}$ 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 derivations of a transition $P \xrightarrow{a,\lambda}_{pepa} P'$ 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 for a subset of processes $C \subseteq \mathcal{P}_{pepa}$ and $a \in \mathcal{A}$ is given by (see, e.g. [Hillston(1996), Hillston(2005)]):

$$q[P,C,a] = \textstyle\sum_{Q \in C} \sum \{\!|\, \lambda \mid P \xrightarrow{a,\lambda}_{pepa} Q \,|\!\}.$$

Here, $\{\!|\, P \xrightarrow{a,\lambda}_{pepa} Q \,|\!\}$ is the multiset of transitions $P \xrightarrow{a,\lambda}_{pepa} Q$ and $\{\!|\, \lambda \mid P \xrightarrow{a,\lambda}_{pepa} Q \,|\!\}$ is the multiset of all λ's involved. The multiplicity of $P \xrightarrow{a,\lambda}_{pepa} Q$ is to be interpreted as the number of different ways the transition can be derived using the rules of Figure 11. We are now ready to define *PEPA*'s notion of strong equivalence[1] [Hillston(1996), Hillston(2005)].

Definition 9 *An equivalence relation $R \subseteq \mathcal{P}_{pepa} \times \mathcal{P}_{pepa}$ is called a strong equivalence if*

$$q[P_1,[Q]_R,a] = q[P_2,[Q]_R,a]$$

for all $P_1, P_2 \in \mathcal{P}_{pepa}$ such that $R(P_1,P_2)$, all $Q \in \mathcal{P}_{pepa}$ and all $a \in \mathcal{A}$. Two processes $P_1, P_2 \in \mathcal{P}_{pepa}$ are strongly equivalent if $R(P_1,P_2)$ for a strong equivalence R, notation $P_1 \sim_{pepa} P_2$. •

The next lemma couples, for a *PEPA*-process P, an action a and a function $\mathscr{P} \in \mathcal{FS}(\mathcal{P}_{pepa}, \mathbb{R}_{\geqslant 0})$, the evaluation $\mathscr{P}(P')$ 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.

[1] In [Hillston(1996)] strong equivalence is denoted by $\cong$; in this paper, we use $\sim_{pepa}$, instead, for notational uniformity.

Lemma 14 *Let $P \in \mathcal{P}_{pepa}$ and $a \in \mathcal{A}$. Suppose $P \overset{\delta_a}{\rightarrowtail}_{pepa} \mathscr{P}$. The following holds: $\mathscr{P}(P') = \sum \{\!|\, \lambda \mid P \xrightarrow{a,\lambda}_{pepa} P' \,|\!\}$ for all $P' \in \mathcal{P}_{pepa}$.*

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

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

$$
\begin{array}{lll}
\multicolumn{2}{l}{\sum \{\!|\, \lambda \mid P_1 \bowtie_A P_2 \xrightarrow{a,\lambda}_{pepa} P' \,|\!\}} & \\
= & \sum \{\!|\, \lambda \mid P_1 \xrightarrow{a,\lambda}_{pepa} P_1' \,|\!\} & \text{by rule (PAR1a)} \\
= & \mathscr{P}_1(P_1') & \text{by the induction hypothesis} \\
= & \mathscr{P}_1(P_1') \cdot \mathbf{D}_{P_2}(P_2) & \text{since } \mathbf{D}_{P_2}(P_2) = 1 \\
= & (\mathscr{P}_1 \bowtie_A \mathbf{D}_{P_2})(P_1' \bowtie_A P_2) + & \\
 & (\mathbf{D}_{P_1} \bowtie_A \mathscr{P}_2)(P_1' \bowtie_A P_2) & \text{definition } \bowtie_A \text{ on } \mathcal{FS}(\mathcal{P}_{pepa}, \mathbb{R}_{\geqslant 0}),\ \mathbf{D}_{P_1}(P_1') = 0 \\
= & \mathscr{P}(P') & \text{by rule (PAR1)}
\end{array}
$$

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

$$
\begin{array}{lll}
\multicolumn{2}{l}{\sum \{\!|\, \lambda \mid P_1 \bowtie_A P_2 \xrightarrow{a,\lambda}_{pepa} P' \,|\!\}} & \\
= & (\sum \{\!|\, \lambda \mid P_1 \xrightarrow{a,\lambda}_{pepa} P_1 \,|\!\}) + & \\
 & (\sum \{\!|\, \lambda \mid P_2 \xrightarrow{a,\lambda}_{pepa} P_2 \,|\!\}) & \text{by rules (PAR1a) and (PAR1b)} \\
= & \mathscr{P}_1(P_1) + \mathscr{P}_2(P_2) & \text{by the induction hypothesis} \\
= & (\mathscr{P}_1 \bowtie_A \mathbf{D}_{P_2})(P_1 \bowtie_A P_2) + & \\
 & (\mathbf{D}_{P_1} \bowtie_A \mathscr{P}_2)(P_1 \bowtie_A P_2) & \text{definition } \bowtie_A \text{ on } \mathcal{FS}(\mathcal{P}_{pepa}, \mathbb{R}_{\geqslant 0}),\ \mathbf{D}_{P_1}(P_1),\ \mathbf{D}_{P_2}(P_2) = 1 \\
= & \mathscr{P}(P') & \text{again by rule (PAR1)}
\end{array}
$$

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

$$
\begin{array}{lll}
\multicolumn{2}{l}{\sum \{\!|\, \lambda \mid P_1 \bowtie_A P_2 \xrightarrow{a,\lambda}_{pepa} P' \,|\!\}} & \\
= & \sum \{\!|\, arf(P_1, P_2) \cdot \lambda_1 \cdot \lambda_2 \mid P_1 \xrightarrow{a,\lambda_1}_{pepa} P_1',\ P_2 \xrightarrow{a,\lambda_2}_{pepa} P_2' \,|\!\} & \text{by rule (PAR2)} \\
= & arf(P_1, P_2) \cdot & \\
 & \left(\sum \{\!|\, \lambda_1 \mid P_1 \xrightarrow{a,\lambda_1}_{pepa} P_1' \,|\!\} \right) \cdot \left(\sum \{\!|\, \lambda_2 \mid P_2 \xrightarrow{a,\lambda_2}_{pepa} P_2' \,|\!\} \right) & \text{by distributivity} \\
= & arf(P_1, P_2) \cdot \mathscr{P}_1(P_1') \cdot \mathscr{P}_2(P_2') & \text{by the induction hypothesis} \\
= & arf(\mathscr{P}_1, \mathscr{P}_2) \cdot \mathscr{P}_1(P_1') \cdot \mathscr{P}_2(P_2') & \text{by Lemma 12} \\
= & arf(\mathscr{P}_1, \mathscr{P}_2) \cdot (\mathscr{P}_1 \bowtie_A \mathscr{P}_2)(P_1' \bowtie_A P_2') & \text{definition } \bowtie_A \text{ on } \mathcal{FS}(\mathcal{P}_{pepa}, \mathbb{R}_{\geqslant 0}) \\
= & \mathscr{P}(P') & \text{by rule (PAR2)}
\end{array}
$$

The other cases are simpler and omitted here.
With the lemma in place we can prove the following correspondence result for $\mathcal{S}_{pepa}$-bisimilarity and strong equivalence as given by Definition 9.

Theorem 15 *For any two PEPA-processes $P_1, P_2 \in \mathcal{P}_{pepa}$ the following holds: $P_1 \simeq_{\mathcal{S}_{pepa}} P_2$ iff $P_1 \sim_{pepa} P_2$.*

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

$$\begin{array}{rcll} q[P, [Q]_R, a] & = & \sum_{Q' \in [Q]_R} \sum \{\!|\, \lambda \mid P \xrightarrow{a,\lambda}_{pepa} Q' \,|\!\} & \text{by definition } q[P, [Q]_R, a] \\ & = & \sum_{Q' \in [Q]_R} \mathscr{P}(Q') & \text{by Lemma 14} \\ & = & \sum_{Q' \in [Q]_R} \theta_{pepa}(P)(a)(Q') & \text{by definition } \theta_{pepa} \end{array}$$

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 $Q \in \mathcal{P}_{pepa}$, $a \in \mathcal{A}$ iff $\sum_{Q' \in [Q]_R} \theta_{pepa}(P_1)(a)(Q') = \sum_{Q' \in [Q]_R} \theta_{pepa}(P_2)(a)(Q')$ for all $Q \in \mathcal{P}_{pepa}$, $a \in \mathcal{A}$. Thus, the equivalence relation R is a strong equivalence iff R is an $\mathcal{S}_{pepa}$-bisimulation, from which the theorem follows.
In view of our general correspondence result Theorem 7, the above theorem shows that *PEPA*'s strong equivalence $\sim_{pepa}$ is a behavioural equivalence, viz. the behavioural equivalence on the *FuTS* $\mathcal{S}_{pepa}$, when seen as a $\mathcal{V}^{\Delta_{\mathcal{A}}}_{\mathbb{R}_{\geqslant 0}}$-coalgebra, which, in turn, coincides with the associated coalgebraic bisimilarity. In other words, letting $\mathcal{V}_{pepa}$ abbreviate $\mathcal{V}^{\Delta_{\mathcal{A}}}_{\mathbb{R}_{\geqslant 0}}$, the following equalities hold:

$$\sim_{pepa} \;=\; \simeq_{\mathcal{S}_{pepa}} \;=\; \approx^{\mathcal{S}_{pepa}}_{\mathcal{V}_{pepa}} \;=\; \sim^{\mathcal{S}_{pepa}}_{\mathcal{V}_{pepa}}$$

Thus, PEPA's standard, *FuTS*, behavioural and coalgebraic semantics coincide.

7 *FuTS* Semantics of *IML*

In this section we provide a *FuTS* semantics for a relevant part of *IML*, the language of Interactive Markov Chains [Hermanns(2002)], *IMC*s for short. *IMC*s 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 distributions. As a consequence, *IMC*s embody both non-deterministic and stochastic behaviour. System analysis using *IMC*s proves to be a powerful approach because of the orthogonality of qualitative and quantitative dynamics, their logical underpinning and tool support, cf. [Bohnenkamp et al.(2006), Hermanns & Katoen(2010)] and [Bozga et al.(2012)]. A number of behavioural equivalences, both strong and weak, are available for *IMC*s [Eisentraut et al.(2010a)]. In our treatment here, discussing a subset we call *IML*s, we do not deal with internal τ-steps and focus on strong bisimilarity. The *FuTS* semantics we consider in the sequel has been proposed in [De Nicola et al.(2011)].

Definition 10 *The set $\mathcal{P}_{iml}$ of IML processes is given by the grammar below*

$$P \;::=\; \mathbf{nil} \mid a.P \mid \lambda.P \mid P + P \mid P \parallel_A P \mid X$$

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 $\mathcal{X}$. •

We assume the same notation and guardedness requirements for constant definition and usage as in Section 6 for *PEPA*.

In line with the discussion above, 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 $a.\lambda.\mathbf{nil} + \mu.b.\mathbf{nil}$ in *IML*. It should be noted that for *IMC*s actions are considered to take no time.

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

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

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

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

Figure 12: *FuTS* semantics for *IML*.

Definition 11 *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 12.* •

To accommodate for action-based and delay-related transitions, the *FuTS* $\mathcal{S}_{iml}$ is non-simple, having the two state-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) 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 parallel construct $\parallel_A$, interleaving applies both for non-synchronised 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 $\rightarrowtail_1$. In rule (PAR1), for clarity, we decorated the characteristic functions, writing $\mathbf{D}^i_P$, for $i = 1, 2$, for $\mathbf{D}_P = [\, P \mapsto \mathtt{true}\,]$ in $\mathcal{FS}(\mathcal{P}_{iml}, \mathbb{B})$ and $\mathbf{D}_P = [\, P \mapsto 1\,]$ in $\mathcal{FS}(\mathcal{P}_{iml}, \mathbb{R}_{\geqslant 0})$. We recall that for all $R \in \mathcal{P}_{iml}$:

$$(\mathscr{P} \parallel_A \mathscr{Q})(R) = \begin{cases} \mathscr{P}(R_1) \cdot \mathscr{Q}(R_2), \text{if } R = R_1 \parallel_A R_2 \text{ for some } R_1, R_2 \in \mathcal{P}_{iml} \\ 0, \text{otherwise} \end{cases}$$

where $\cdot$ is the product in $\mathbb{R}_{\geqslant 0}$ whenever $\mathscr{P}, \mathscr{Q} \in \mathcal{FS}(\mathcal{P}_{iml}, \mathbb{R}_{\geqslant 0})$ and is the logical conjunction $\wedge$ for $\mathscr{P}, \mathscr{Q} \in \mathcal{FS}(\mathcal{P}_{iml}, \mathbb{B})$.

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

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

$$\lambda.b.X \parallel_A \mu.b.Y \overset{\delta}{\rightarrowtail}_2 [b.X \parallel_A \mu.b.Y \mapsto \lambda,\ \lambda.b.X \parallel_A b.Y \mapsto \mu\,]$$

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 $\simeq_{\mathcal{S}_{iml}}$ for the associated bisimilarity.

Lemma 16 *The FuTS $\mathcal{S}_{iml}$ is total and deterministic.* □

The standard SOS semantics of *IML* [Hermanns(2002)] is given in Figure 13 involving the transition relations

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

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

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

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

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

Figure 13: Standard SOS rules for *IML*.

Below we will use the functions $\mathbf{T}$ and $\mathbf{R}$ based on $\rightarrow$ and $\dashrightarrow$, cf. [Hermanns & Katoen(2010)]. We have $\mathbf{T}\colon \mathcal{P}_{iml} \times \mathcal{A} \times \mathbf{2}^{\mathcal{P}_{iml}} \to \mathbb{B}$ given by $\mathbf{T}(P, a, C) = \mathtt{true}$ 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. Alternatively, one could define an explicit *cnt*-function, $cnt : \mathcal{P}_{iml} \times \mathbb{R}_{>0} \times \mathcal{P}_{iml} \to \mathbb{R}_{\geqslant 0}$ returning the number of multiplicities of a transition $P \overset{\lambda}{\dashrightarrow} P'$. 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 $P \in \mathcal{P}_{iml}$, $C \subseteq \mathcal{P}_{iml}$. For *IML* we have the following notion of strong bisimulation [Hermanns(2002), Hermanns & Katoen(2010)] that we will compare with the notion of bisimulation associated with the *FuTS* $\mathcal{S}_{iml}$.

Definition 12 *An equivalence relation $R \subseteq \mathcal{P}_{iml} \times \mathcal{P}_{iml}$ is called a strong bisimulation for IML if, for all $P_1, P_2 \in \mathcal{P}_{iml}$ such that $R(P_1, P_2)$, it holds that*

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

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

To establish the correspondence of *FuTS* bisimilarity $\simeq_{\mathcal{S}_{iml}}$ for $\mathcal{S}_{iml}$ of Definition 11 and strong bisimilarity $\sim_{iml}$ 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 17 *(a) Let $P \in \mathcal{P}_{iml}$ and $a \in \mathcal{A}$. If $P \overset{a}{\rightarrowtail}_1 \mathscr{P}$ then $P \xrightarrow{a} P' \iff \mathscr{P}(P') = \mathtt{true}$.*

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

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 $\lambda.P \overset{a}{\rightarrowtail}_1 \mathscr{P}$. Then we have $\mathscr{P} = []_{\mathbb{B}}$. Both $\lambda.P \xrightarrow{a} P'$ for no $P' \in \mathcal{P}_{iml}$, as no transition is provided in $\rightarrow$, and $\mathscr{P}(P') = \mathtt{false}$ by definition of $[]_{\mathbb{B}}$, for all $P' \in \mathcal{P}_{iml}$.

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

$Q' = Q$, for $Q' \in \mathcal{P}_{iml}$. We have

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

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 $P \overset{\delta}{\rightarrowtail}_2 \mathscr{P}$. Suppose $P = \mu.P'$. Then it holds that P admits a single $\dashrightarrow$-transition, viz. $P \overset{\mu}{\dashrightarrow} P'$. Thus we have $\sum \{\!| \, \lambda \mid P \overset{\lambda}{\dashrightarrow} P' \, |\!\} = \mu = [\, P' \mapsto \mu \,](P') = \mathscr{P}(P')$. Suppose $P = \mu.P''$ for some $P'' \neq P$. Then we have $\sum \{\!| \, \lambda \mid P \overset{\lambda}{\dashrightarrow} P' \, |\!\} = 0 = [\, P'' \mapsto \mu \,](P') = \mathscr{P}(P')$.

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

$$
\begin{array}{ll}
& \sum \{\!| \, \lambda \mid P_1 \parallel_A P_2 \overset{\lambda}{\dashrightarrow} P' \, |\!\} \\
= & \sum \{\!| \, \lambda \mid P_1 \overset{\lambda}{\dashrightarrow} P_1',\ P' = P_1' \parallel_A P_2 \, |\!\} + \sum \{\!| \, \lambda \mid P_2 \overset{\lambda}{\dashrightarrow} P_2',\ P' = P_1 \parallel_A P_2' \, |\!\} \\
& \text{by analysis of } \dashrightarrow \\
= & (\, \mathtt{if}\ P' = P_1' \parallel_A P_2\ \mathtt{then} \sum \{\!| \, \lambda \mid P_1 \overset{\lambda}{\dashrightarrow} P_1' \, |\!\}\ \mathtt{else}\ 0\ \mathtt{end} \,) + \\
& \qquad (\, \mathtt{if}\ P' = P_1 \parallel_A P_2'\ \mathtt{then} \sum \{\!| \, \lambda \mid P_2 \overset{\lambda}{\dashrightarrow} P_2' \, |\!\}\ \mathtt{else}\ 0\ \mathtt{end} \,) \\
= & (\, \mathtt{if}\ P' = P_1' \parallel_A P_2\ \mathtt{then}\ \mathscr{P}_1(P_1')\ \mathtt{else}\ 0\ \mathtt{end} \,) + \\
& \qquad (\, \mathtt{if}\ P' = P_1 \parallel_A P_2'\ \mathtt{then}\ \mathscr{P}_2(P_2')\ \mathtt{else}\ 0\ \mathtt{end} \,) \\
& \text{by induction hypothesis for } P_1 \text{ and } P_2 \\
= & (\, \mathscr{P}_1 \parallel_A \mathbf{D}_{P_2} \,)(P') + (\, \mathbf{D}_{P_1} \parallel_A \mathscr{P}_2 \,)(P') \\
& \text{by definition of } \parallel_A, \mathbf{D}_{P_1}, \mathbf{D}_{P_2} \text{ and } + \text{ on } \mathcal{FS}(\mathcal{P}_{iml}, \mathbb{R}_{\geqslant 0}\,) \\
= & \mathscr{P}(P')
\end{array}
$$

The remaining cases are left to the reader.

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

Theorem 18 *For any two processes $P_1, P_2 \in \mathcal{P}_{iml}$ it holds that $P_1 \simeq_{\mathcal{S}_{iml}} P_2$ iff $P_1 \sim_{iml} P_2$.*

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

$$\begin{array}{rcll}
\mathbf{T}(P, a, [Q]_R) & \Leftrightarrow & \exists Q' \in [Q]_R \colon P \xrightarrow{a} Q' & \text{by definition of } \mathbf{T} \\
& \Leftrightarrow & \exists Q' \in [Q]_R \colon \mathscr{P}(Q') = \mathtt{true} & \text{by Lemma 17a} \\
& \Leftrightarrow & \sum_{Q' \in [Q]_R} \theta_1(P)(a)(Q) = \mathtt{true} & \text{by definition of } \theta_1
\end{array}$$

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

$$\begin{array}{rcll}
\mathbf{R}(P, [Q]_R) & = & \sum_{Q' \in [Q]_R} \sum \{\!|\, \lambda \mid P \overset{\lambda}{\dashrightarrow} Q' \,|\!\} & \text{by definition of } \mathbf{R} \\
& = & \sum_{Q' \in [Q]_R} \mathscr{P}(Q') & \text{by Lemma 17b} \\
& = & \sum_{Q' \in [Q]_R} \theta_2(P)(\delta)(Q) & \text{by definition of } \theta_2
\end{array}$$

Combining the equations, we conclude that a strong bisimulation for *IML* is also an $\mathcal{S}_{iml}$-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 bisimilarity $P_1 \sim_{iml} P_2$ is coalgebraically underpinned, as it coincides, calling to Theorem 7 once more, with behavioural equivalence on the *FuTS* $\mathcal{S}_{iml}$, when seen as a $\mathcal{V}^{\langle \mathcal{A}, \Delta \rangle}_{\langle \mathbb{B}, \mathbb{R}_{\geqslant 0} \rangle}$-coalgebra, which, in turn, coincides with the associated coalgebraic bisimilarity. In other words, letting $\mathcal{V}_{iml}$ abbreviate $\mathcal{V}^{\langle \mathcal{A}, \Delta \rangle}_{\langle \mathbb{B}, \mathbb{R}_{\geqslant 0} \rangle}$, the following equalities hold:

$$\sim_{iml} \;=\; \simeq_{\mathcal{S}_{iml}} \;=\; \approx^{\mathcal{S}_{iml}}_{\mathcal{V}_{iml}} \;=\; \sim^{\mathcal{S}_{iml}}_{\mathcal{V}_{iml}}$$

Thus, IML's standard, *FuTS*, behavioural and coalgebraic semantics coincide.

8 Discussion

Above we have focused on the treatment with *FuTS*s of action prefix in the setting of an elementary process language, stochastic prefix in the setting of *PEPA* and their mixing in the setting of *IML* and studied the positioning of the associated notion of process strong bisimulation equivalences. The semirings involved are the booleans $\mathbb{B}$ and non-negative reals $\mathbb{R}_{\geqslant 0}$. The orthogonality of *FuTS*s allows for superposition of various state-to-function transition relations, allowing e.g. to mingle with discrete deterministic time, as we will sketch below. Also, when considering repeated application of functors $\mathcal{FS}(\cdot, \mathcal{R})$ more complex state-to-function transition system can be defined, for example to deal with so-called Markov automata [Eisentraut et al.(2010a), Eisentraut et al.(2010b)].

With the help of the semiring of non-negative integers $\mathbb{N}$, a *FuTS*-style semantics can be given to discrete deterministic time processes where time elapses by the ticking of the clock. For example, [Aldini et al.(2010)] discusses a small process language, called *TPC*, involving the prefix construct $(n).P$, with $n \in \mathbb{N}$, $n > 0$, expressing that the process P is to be executed after n time steps. A *FuTS* for *TPC* can be of type $(\mathcal{P}_{TPC}, \rightarrowtail_1, \rightarrowtail_2)$ over the label sets $\mathcal{A}$ and $\{\surd\}$—where $\mathcal{A}$ is a set of actions as before and the special symbol $\surd$ denotes a fixed discrete deterministic delay representing progress in time—and the semirings $\mathbb{B}$ and $\mathbb{N}$. We have $\rightarrowtail_1 \subseteq \mathcal{P}_{TPC} \times \mathcal{A} \times \mathcal{FS}(\mathcal{P}_{TPC}, \mathbb{B})$ and $\rightarrowtail_2 \subseteq \mathcal{P}_{TPC} \times \{\surd\} \times \mathcal{FS}(\mathcal{P}_{TPC}, \mathbb{N})$. The relevant rules involving the timed prefix construct are

$$\text{(TPF1)}\ \frac{a \in \mathcal{A}}{(n).P \overset{a}{\rightarrowtail}_1 []_{\mathbb{B}}} \qquad \text{(TPF2)}\ \frac{P \overset{\surd}{\rightarrowtail}_2 \mathscr{P}}{(n).P \overset{\surd}{\rightarrowtail}_2 [n; P] + [P \mapsto n] + (n + \mathscr{P})}$$

The first timed prefix rule (TPF1) expresses that a timed prefix cannot perform an action (immediately). The second time prefix rule (TPF2) combines a possible evolution over time of the process P

into its continuation $\mathscr{P}$ with the elapse of the prefix. Note, the continuation in the conclusion of rule (TPF2) is a sum of three parts, viz. $[n; P]$, $[\, P \mapsto n\,]$, $(n + \mathscr{P})$. The mappings $[n; P]$ and $(n + \mathscr{P})$ are given by

$$[n; P](Q) = \begin{cases} m & \text{if } 0 < m < n \text{ and } Q = (n-m).P \\ 0 & \text{otherwise} \end{cases} \qquad (n + \mathscr{P})(Q) = \begin{cases} n + \mathscr{P}(Q) & \text{if } \mathscr{P}(Q) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Time progress taking fewer steps than n is covered by the continuation $[n; P]$. For m strictly between 0 and n, after m time steps there remains $(n - m).P$ to be executed. After exactly n time steps, P is to be executed. After more than n time steps, say $n + m$ time steps, process Q is to be executed if $\mathscr{P}(Q) = m$, for $m > 0$.

The rules for the choice and parallel construct of *TPC* make use of corresponding operations on $\mathcal{FS}(\mathcal{P}_{TPC}, \mathbb{N})$ such that so-called time-determinism and time-continuity principles are respected. Again a total and deterministic *FuTS* is obtained this way. Also, along the lines of the correspondence proofs for *PEPA* and *IML*, it can be shown that the notion of discrete-time bisimulation of [Aldini et al.(2010), Baeten & Middelburg(2002)] and the notion of bisimulation for the *FuTS* sketched above as well as the associated notion of behavioural equivalence coincide. Thus, also deterministic time can be handled with *FuTS*s. Note, as the semiring $\mathbb{N}$ does not possess multiplicative inverses, we cannot appeal to Theorem 9 to connect to coalgebraic equivalence in this case.

Markov automata, as proposed in [Eisentraut et al.(2010a), Eisentraut et al.(2010b)], combine non-deterministic and probabilistic behaviour with stochastic time. The combination of non-deterministic and probabilistic behaviour provided by Markov automata can be easily achieved, at the linguistic level by means of a combination of a standard choice operator, $+$, with the following probabilistic extension of action prefix: $a.\{p_1 :: P_1 \Box \ldots \Box p_h :: P_h\}$ with $a \in \mathcal{A}$, the set of actions, and $h > 0$, $p_1, \ldots, p_h \in (0, 1]$ such that $p_1 + \cdots + p_h \leqslant 1$. The syntactic construct $\{p_1 :: P_1 \Box \ldots \Box p_h :: P_h\}$ denotes the sub-distribution $\mathcal{D}_{\{p_1::P_1\Box\ldots\Box p_h::P_h\})}$ over processes defined by

$$\mathcal{D}_{\{p_1::P_1\Box\ldots\Box p_h::P_h\}} = \textstyle\sum_{i=1}^{h} [P_i \mapsto p_i]$$

The intuitive meaning is then obvious: process $a.\{p_1 :: P_1 \Box \ldots \Box p_h :: P_h\}$ performs action a and then behaves as process P with probability $\mathcal{D}_{\{p_1::P_1\Box\ldots\Box p_h::P_h\}}(P)$.

The amalgamation of action prefix and probabilistic choice leads to a nesting of the functors involved. Now, transitions labelled by actions go from processes to *sets of discrete sub-distributions*. In fact, following the *FuTS* approach, one encounters transitions of the form $\rightarrowtail_1 \subseteq \mathcal{P}_{MA} \times \mathcal{A} \times \mathcal{FS}(\mathit{SDistr}(\mathcal{P}_{MA}), \mathbb{B})$ to deal with the non-deterministic/probabilistic aspect of the language as well as transitions of the form $\rightarrowtail_2 \subseteq \mathcal{P}_{MA} \times \Delta \times \mathcal{FS}(\mathcal{P}_{MA}, \mathbb{R}_{\geqslant 0})$ to handle stochastic delays, as we have already seen in the previous section. Here, $\mathit{SDistr}(\cdot)$ is the functor associating finite sub-distributions to a set, dealing with functions similar to the *FuTS* functors. More concretely, for the combined action and probabilistic choice prefix, we have the rules

$$(\text{APF1})\ \frac{}{a.\{p_1 :: P_1 \Box \ldots \Box p_h :: P_h\} \overset{a}{\rightarrowtail}_1 [\mathcal{D}_{(\{p_1::P_1\Box\ldots\Box p_h::P_h\})} \mapsto \mathtt{true}]}$$

$$(\text{APF2})\ \frac{b \neq a}{a.\{p_1 :: P_1 \Box \ldots \Box p_h :: P_h\} \overset{b}{\rightarrowtail}_1 []_{\mathbb{B}}} \qquad (\text{APF3})\ \frac{}{a.\{p_1 :: P_1 \Box \ldots \Box p_h :: P_h\} \overset{\delta}{\rightarrowtail}_2 []_{\mathbb{R}_{\geqslant 0}}}$$

We expect that all coalgebraic reasoning will hold true for such nesting of functors. In particular, we claim for a generalised notion of (deterministic and total) *FuTS*, viz. coalgebras (X, θ) of a functor $(\mathcal{F}_1 \circ \cdots \circ \mathcal{F}_n)^{\mathcal{L}}$ or a product of such functors, where $\mathcal{F}_i = \mathcal{FS}(\cdot, \mathcal{R}_i)$ for some semiring $\mathcal{R}_i$, $i = 1 \ldots n.$, that the associated notion of *FuTS* bisimilarity coincides with behavioural equivalence, and, for semirings having multiplicative inverses and meeting the zero-sum property (as discussed in Section 4), with coalgebraic bisimilarity as well.

9 Concluding remarks

Total and deterministic state-to-function labeled transition systems, *FuTS*s, are a convenient instrument to express the operational semantics of both qualitative and quantitative process languages. In this paper we have discussed the notion of bisimilarity that arises from a *FuTS*, possibly involving multiple transition relations, from a coalgebraic perspective. For *FuTS* models of two process languages based on prominent stochastic process algebras we related the induced notion of bisimulation to the standard equivalences, thus providing these equivalence with a coalgebraic underpinning. The main technical contribution of our paper is a correspondence result, Theorem 7, that relates bisimilarity of a *FuTS* $\mathcal{S}$ to behavioural equivalence of the functor associated with $\mathcal{S}$.

It is noted in [Bonchi et al.(2011)], 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 bisimilarity and of behavioural equivalence may not coincide. A counter example is provided, cf. [Bonchi et al.(2011), Section 2.2]. Essential for the construction of the counter-example, 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 [de Vink & Rutten(1999)], for coalgebraic bisimilarity and *FuTS* bisimilarity 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 [Klin & Sassone(2008)], although no detailed proof is provided there, this will not happen with $\mathbb{R}_{\geqslant 0}$ as underlying semiring. Here we propose to consider semirings which admit a (right) multiplicative inverse for non-zero elements, and satisfy the so-called zero-sum property, stating 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 have shown, Theorem 9, that, when the semirings involved enjoy these properties, weak pullbacks are preserved by the associated functor. Therefore, coalgebraic bisimilarity and behavioural equivalence are the same. As a consequence, under conditions which are met by the *SPC*s proposed in the literature, we have that *FuTS*-bisimilarity, behavioural equivalence and coalgebraic bisimilarity coincide.

For two prototypical stochastic process languages based on *PEPA* and on *IMC* we have shown that the notion of strong equivalence and strong bisimilarity associated with these calculi, coincides with the notion of bisimilarity of the corresponding *FuTS*. Using these *FuTS*s as a stepping stone, the correspondence result bridges between the concrete notion of bisimulation for *PEPA* and *IML*, and the coalgebraic notions of behavioural equivalence and coalgebraic bisimilarity. Hence, from this perspective, the concrete notions are seen as the natural strong equivalence to consider. Obviously, classical strong bisimilarity [Milner(1980), Park(1981)] and bisimilarity for *FuTS* over $\mathbb{B}$ coincide. Also, strong bisimulation of [Hillston(1996)] 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. Finally, we gave an account how languages based on discrete deterministic time as well as those where stochastic time is integrated with discrete probability and with non-determinism can be easily treated in the *FuTS* framework. 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 behavioural equivalence will amount to similar identifications also for the above mentioned models. In particular, the study of nested functors (i.e. compositions of functors) seems to be promising.

Acknowledgments The authors are grateful to Rocco De Nicola, Fabio Gadducci, Daniel Gebler, Michele Loreti and Jan Rutten for fruitful discussions on the subject and useful suggestions. DL and MM acknowledge support by EU Project n. 257414 Autonomic Service-Components Ensembles (ASCENS). 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

[Aldini et al.(2010)] Aldini, A., Bernardo, M., & Corradini, F. (2010). *A Process Algebraic Approach to Software Architecture design*. Springer.

[Baeten & Middelburg(2002)] Baeten, J. & Middelburg, C. (2002). *Process Algebra with Timing*. Springer.

[Baeten et al.(2009)] Baeten, J., Basten, T., & Reniers, M. (2009). *Process Algebra: Equational Theories of Communicating Processes*, volume 50 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press.

[Baier et al.(2004)] Baier, C., Haverkort, B., Hermanns, H., Katoen, J.-P., & Siegle, M., editors (2004). *Validation of Stochastic Systems – A Guide to Current Research*. LNCS 2925.

[Barr(1993)] Barr, M. (1993). Terminal coalgebras in well-founded set theory. *Theoretical Computer Science*, **221**, 299–315.

[Bernardo(2007)] Bernardo, M. (2007). A survey of markovian behavioral equivalences. In M. Bernardo and J. Hillston, editors, *SFM 2007 Advanced Lectures*, pages 180–219. LNCS 4486.

[Bernardo & Gorrieri(1998)] Bernardo, M. & Gorrieri, R. (1998). A tutorial on EMPA: a theory of concurrent processes with non-determinism, priorities, probabilities and time. *Theoretical Computer Science*, **202**(1–2), 1–54.

[Bohnenkamp et al.(2006)] Bohnenkamp, H., D'Argenio, P., Hermanns, H., & Katoen, J.-P. (2006). MODEST: A compositional modeling formalism for hard and softly timed systems. *IEEE Transactions on Software Engineering*, **32**(10), 812–830.

[Bonchi et al.(2011)] Bonchi, F., Bonsangue, M., Boreale, M., Rutten, J., & Silva, A. (2011). A coalgebraic perspective on linear weighted automata. Technical Report SEN–1104, CWI. 31pp, to appear in Information and Computation.

[Boreale(2009)] Boreale, M. (2009). Weighted bisimulation in linear algebraic form. In M. Bravetti and G. Zavattaro, editors, *Proc. CONCUR 2009*, pages 163–177. LNCS 5710.

[Boreale & Gadducci(2006)] Boreale, M. & Gadducci, F. (2006). Processes as formal power series: A coinductive approach to denotational semantics. *Theoretical Computer Science*, **360**(1–3), 440–458.

[Bozga et al.(2012)] Bozga, M., David, A., Hartmanns, A., Hermanns, H., Larsen, K., Legay, A., & Tretmans, J. (2012). State-of-the-art tools and techniques for quantitative modeling and analysis of embedded systems. In W. Rosenstiel and L. Thiele, editors, *Proc. DATE 2012*, pages 370–375. IEEE.

[Cardelli & Mardare(2010)] Cardelli, L. & Mardare, R. (2010). The measurable space of stochastic processes. In *Proc. QEST 2010, Williamsburg*, pages 171–180. IEEE Computer Society.

[de Bakker & de Vink(1996)] de Bakker, J. & de Vink, E. (1996). *Control Flow Semantics*. The MIT Press.

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

[De Nicola et al.(2009)] De Nicola, R., Latella, D., Loreti, M., & Massink, M. (2009). Rate-based transition systems for stochastic process calculi. In S. Albers et al., editor, *Proc. ICALP 2009, Part II*, pages 435–446. LNCS 5556.

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

[de Vink & Rutten(1999)] de Vink, E. & Rutten, J. (1999). Bisimulation for probabilistic transition systems: a coalgebraic approach. *Theoretical Computer Science*, **221**, 271–293.

[Eisentraut et al.(2010a)] Eisentraut, C., Hermanns, H., & Zhang, L. (2010a). Concurrency and composition in a stochastic world. In P. Gastin and F. Laroussinie, editors, *Proc. CONCUR 2010*, pages 21–39. LNCS 6269.

[Eisentraut et al.(2010b)] Eisentraut, C., Hermanns, H., & Zhang, L. (2010b). On probabilistic automata in continuous time. In *Proc. LICS, Edinburgh*, pages 342–351. IEEE Computer Society.

[Gumm & Schröder(2001)] Gumm, H. & Schröder, T. (2001). Monoid-labeled transition systems. *Electronic Notes in Theoretical Computer Science*, **44**(1), 185–204.

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

[Gumm & Schröder(2002)] Gumm, H. & Schröder, T. (2002). Coalgebras of bounded type. *Mathematical Structures in Computer Science*, **12**, 565–578.

[Hermanns(2002)] Hermanns, H. (2002). *Interactive Markov Chains*. LNCS 2428.

[Hermanns & Katoen(2010)] Hermanns, H. & Katoen, J.-P. (2010). The how and why of interactive markov chains. In F. de Boer, M. Bonsangue, S. Hallerstede, and M. Leuschel, editors, *Proc. FMCO 2009*, pages 311–337. LNCS 6286.

[Hermanns et al.(1998)] Hermanns, H., Herzog, U., & Mertsiotakis, V. (1998). Stochastic process algebras – between LOTOS and Markov chains. *Computer Networks and ISDN Systems*, **30**, 901–924.

[Hermanns et al.(2002)] Hermanns, H., Herzog, U., & Katoen, J.-P. (2002). Process algebra for performance evaluation. *Theoretical Computer Science*, **274**(1–2), 43–87.

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

[Hillston(2005)] Hillston, J. (2005). Process algebras for quantitative analysis. In *Proc. LICS, Chicago*, pages 239–248. IEEE.

[Hoare(1985)] Hoare, C. (1985). *Communicating Sequential Processes*. Prentice Hall.

[Klin(2009)] Klin, B. (2009). Structural operational semantics for weighted transition systems. In J. Palsberg, editor, *Semantics and Algebraic Specification*, pages 121–139. LNCS 5700.

[Klin & Sassone(2008)] Klin, B. & Sassone, V. (2008). Structural operational semantics for stochastic process calculi. In R. Amadio, editor, *Proc. FoSSaCS 2008*, pages 428–442. LNCS 4962.

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

[Latella et al.(2012)] Latella, D., Massink, M., & de Vink, E. (2012). Bisimulation of labeled state-to-function transition systems of stochastic process languages. In T. Soboll and U. Golas, editors, *Proc. ACCAT 2012, Tallin. EPTCS 93*, pages 23–43. EPTCS.

[Marti & Venema(2012)] Marti, J. & Venema, Y. (2012). Lax extensions of coalgebra functors. In D. Pattinson and L. Schröder, editors, *Proc. CMCS 2012*. LNCS. To appear.

[Milner(1980)] Milner, R. (1980). *A Calculus of Communicating Systems*. LNCS 92.

[Milner(1989)] Milner, R. (1989). *Communication and Concurreny*. Prentice Hall.

[Moss(1999)] Moss, L. (1999). Coalgebraic logic. *Annals of Pure and Applied Logic*, **96**, 277–317.

[Park(1981)] Park, D. (1981). Concurrency and automata on infinite sequences. In *Proc. GI-Conference 1981, Karlsruhe*, pages 167–183. LNCS 104.

[Priami(1995)] Priami, C. (1995). Stochastic π-calculus. *The Computer Journal*, **38**(7), 578–589.

[Rutten(2000)] Rutten, J. (2000). Universal coalgebra: a theory of systems. *Theoretical Computer Science*, **249**, 3–80.

[Rutten(2003)] Rutten, J. (2003). Behavioural differential equations: a coinductive calculus of streams, automata, and power series. *Theoretical Computer Science*, **308**(1–3), 1–53.

[Silva(2010)] Silva, A. (2010). *Kleene Coalgebra*. Ph.D. thesis, Radboud University Nijmegen.

[Silva et al.(2011)] Silva, A., Bonchi, F., Bonsangue, M., & Rutten, J. (2011). Quantitative kleene coalgebras. *Information and Computation*, **209**(5), 822–846.

[Sokolova(2005)] Sokolova, A. (2005). *Coalgebraic Analysis of Probabilistic Systems*. Ph.D. thesis, Eindhoven University of Technology.

[Sokolova(2011)] Sokolova, A. (2011). Probabilistic systems coalgebraically: a survey. *Theoretical Computer Science*, **412**(38), 5095–5110.

[Turi & Plotkin(1997)] Turi, D. & Plotkin, G. (1997). Towards a mathematical operational semantics. In *Proc. LICS 1997, Warsaw*, pages 280–291. IEEE.

[van Glabbeek et al.(1995)] van Glabbeek, R., Smolka, S., & Steffen, B. (1995). Reactive, generative and stratified models of probabilistic processes. *Information and Computation*, **121**(1), 59–80.

[Viglizzo(2005)] Viglizzo, I. (2005). Final sequences and final coalgebras for measurable spaces. In J. Fiadeiro, , N. Harman, M. Roggenbach, and J. Rutten, editors, *Proc. CALCO 2005*, pages 395–407. LNCS 3629.

[Wolter(2002)] Wolter, U. (2002). CSP, partial automata, and coalgebras. *Theoretical Computer Science*, **280**, 3–34.