BISIMULATION OF LABELLED STATE-TO-FUNCTION TRANSITION SYSTEMS COALGEBRAICALLY

D. LATELLA, M. MASSINK, AND E.P. DE VINK

CNR – Istituto di Scienza e Tecnologie dell'Informazione 'A. Faedo', Pisa
e-mail address: Diego.Latella@isti.cnr.it

CNR – Istituto di Scienza e Tecnologie dell'Informazione 'A. Faedo', Pisa
e-mail address: Mieke.Massink@isti.cnr.it

Department of Mathematics and Computer Science, Eindhoven University of Technology
Centrum voor Wiskunde en Informatica, Amsterdam
e-mail address: evink@win.tue.nl

Abstract. Labeled state-to-function transition systems, *FuTS* for short, are characterized by transitions which relate states to *functions* of states over general semirings, equipped with a rich set of higher-order operators. As such, *FuTS* constitute a convenient modeling instrument to deal with process languages and their quantitative 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. As generic examples, the equivalences underlying substantial fragments of major examples of quantitative process algebras are related to the bisimilarity of specific *FuTS*. The examples range from a stochastic process language, *PEPA*, to a language for Interactive Markov Chains, *IML*, a (discrete) timed process language, *TPC*, and a language for Markov Automata, *MAL*. The equivalences underlying these languages are related to the bisimilarity of their specific *FuTS*. By the correspondence result coalgebraic justification of the equivalences of these calculi is obtained. The specific selection of languages, besides covering a large variety of process interaction models and modelling choices involving quantities, allows us to show different classes of *FuTS*, namely so-called simple *FuTS*, combined *FuTS*, nested *FuTS*, and general *FuTS*.

1. Introduction

In the last couple of decades, qualitative process languages have been enriched with quantitative information. In the qualitative case, process languages equipped with formal operational semantics have proven to be successful formalisms for the modeling of concurrent systems and the analysis of their behaviour. Generally, the operational semantics of a qualitative process language are given by means of a labeled transition system (*LTS*), with states being process terms and actions decorating the transitions between states. Typically, based on the induced transition system relation, a

1998 ACM Subject Classification: D.2.4 Formal Methods, F.3.1 Specifying and Verifying and Reasoning about Programs, F.3.2 Semantics of Programming Languages .

Key words and phrases: quantitative process algebra, FuTS, function of finite support, bisimulation, coalgebra, behavioral equivalence .

LOGICAL METHODS
IN COMPUTER SCIENCE

DOI:10.2168/LMCS-???

notion of process equivalence is defined, providing means to compare systems and to reduce their representation to enhance subsequent verification.

Extensions of qualitative process languages allow a deterministic as well as stochastic representation of time, or the use of discrete probability distributions for resolving (some) forms of non-determinism. Among them, languages based on stochastic modeling of action duration or delays, usually referred to as stochastic process algebras, or stochastic process calculi (*SPC*), are one of the quantitative enrichments of process languages that have received particular attention. For *SPC*, the main aim has been the integration of qualitative descriptions with quantitative ones in a single mathematical framework, building on the combination of *LTS* and continuous-time Markov chains (CTMC). The latter is one of the most successful approaches to modeling and performance analysis of (computer) systems and networks. An overview of *SPC*, equivalences and related analysis techniques may, for example, be found in [28, 5, 6]. A common feature of many *SPC* is that actions are augmented with the rates of exponentially distributed random variables that characterize their *duration*. Alternatively, actions are assumed to be instantaneous, in which case random variables are used for modeling *delays*, as in [27]. 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 modeling, by means of the choice operator, of the race condition arising from the CTMC interpretation of process behaviour, 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 [31, 27], to proved transition systems [45], to *LTS* with numbered transitions [24, 28], and to unique rate names [19], just to mention a few.

In [15, 17, 16], Latella, Massink et al. proposed a variant of *LTS*, called Rate Transition Systems (*RTS*). In *LTS*, a transition is a triple (P, α, P') where P and α are the source state and the label of the transition, respectively, while P' is the target state reached from P via a transition labeled with α. In *RTS*, 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*, while the third component $\mathscr{P}$ is a *continuation function* (or simply a *continuation* in the sequel), which associates a non-negative real value with each state P'. A non-zero value for state P' represents the rate of the exponential distribution characterizing the average 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. The use of continuations provides a clean and simple solution to the transition multiplicity problem and make *RTS* particularly suited for *SPC* semantics. In order to provide a uniform account of the many *SPC* proposed in the literature, in previous joint work of the first two authors, see [18], State-to-Function Labeled Transition Systems (*FuTS*) have been introduced as a natural generalization of *RTS*. In *FuTS* the codomain of the continuations 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* are equipped with a rich set of (generic) operations on continuation functions, which makes the framework very well suited for a *compositional* definition of the operational semantics of process calculi, including *SPC* and models where both non-deterministic behaviour and stochastic delays are model led, like in the Language of Interactive Markov Chains [27], or even in combination with probabilistic distributions over behaviours, as in languages for Markov Automata [53], besides calculi for deterministic (discrete) timed systems [3].

In this paper we extend the work presented in [18] in two directions. The first contribution concerns the extension of the *FuTS* framework by introducing the notions of combined *FuTS* and

nested *FuTS*. Given label sets $\mathcal{L}_i$ and semirings $\mathcal{R}_i$, a combined *FuTS* takes the general format $\mathcal{S} = (S, \langle \rightarrowtail_i \rangle_{i=1}^n)$ with transition relations $\rightarrowtail_i \subseteq S \times \mathcal{L}_i \times \mathcal{FS}(S, \mathcal{R}_i)$. In the degenerated case of $n = 1$, we speak of a simple *FuTS*, which coincides with the definition of *FuTS* proposed in [18]. Here, $\mathcal{FS}(S, \mathcal{R}_i)$ is the set of total functions from S to $\mathcal{R}_i$ with finite support, a sub-collection of functions also occurring in other work combining coalgebra and quantitative modeling (see, e.g. [35, 10]). So, a combined *FuTS* is characterized by the presence of *multiple* transition relations which allow for a clean definition of the *FuTS* semantics of languages which integrate different aspects of behaviour, such as non-determinism vs. stochastic time, as is characteristic for Interactive Markov Chains. Using a single transition relation in such a setting requires additional proof obligations ensuring type correctness of transition elements, in particular the continuations, as can be seen in [18], for example. Instead, for combined *FuTS* this is ensured by construction. The general format of a so-called nested *FuTS* over the label set $\mathcal{L}$ and semirings $\mathcal{R}_1, \ldots \mathcal{R}_n$, for $n > 1$, is a tuple $\mathcal{S} = (S, \rightarrowtail)$ with $\rightarrowtail \subseteq S \times \mathcal{L} \times \mathcal{FS}((\ldots \mathcal{FS}(S, \mathcal{R}_1) \ldots), \mathcal{R}_n)$. For the purposes of the present paper, $n = 2$ suffices; the nested *FuTS* we consider here are of the form $\mathcal{S} = (S, \rightarrowtail)$ with $\rightarrowtail \subseteq S \times \mathcal{L} \times \mathcal{FS}(\mathcal{FS}(S, \mathcal{R}_1), \mathcal{R}_2)$. For nested *FuTS* the transition relation relates *functions over states*, instead of just states, to continuations. This makes it easy, for instance, to represent non-deterministic choices between probabilistic distributions over behaviours, as it is the case for (the non-timed fragment of languages for) Markov Automata. Finally, product construction for combined *FuTS* and sequencing construction for nested *FuTS*) can easily be combined giving rise to what one may call *general FuTS* (or just *FuTS*, in the sequel), which prove useful for a concise definition of the operational semantics of Markov Automata languages.

We will briefly show how the various types of *FuTS* can be used conveniently for a clean and compact definition of the fragments of interest of major process languages (more details on this can be found in [18], which the interested reader is referred to). For combined *FuTS*, as well as nested *FuTS* and general *FuTS*, we also present *FuTS* bisimilarity, a general notion of bisimilarity, which will also be shown to coincide with the standard bisimilarity of the relevant process languages.

The second direction of investigation presented in this paper consists of a coalgebraic treatment of the various type of *FuTS*. We will see that a combined *FuTS* $(S, \langle \rightarrowtail_i \rangle_{i=1}^n)$ 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 modeling as a function. 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 state 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*. For example, by using Boolean functions, we can model non-deterministic behaviour, as done in Section 7. Similarly, we see that a (two-level) nested *FuTS* $(S, \rightarrowtail)$ is a coalgebra of functor $\mathcal{FS}(\mathcal{FS}(\cdot, \mathcal{R}_1), \mathcal{R}_2)^{\mathcal{L}}$.

Next, the notion of $\mathcal{S}$-bisimilarity that arises from a *FuTS* $\mathcal{S}$ is compared to the coalgebraic notion of behavioral equivalence. Following a familiar argument, we first establish that the functor associated with a *FuTS* $\mathcal{S}$ possesses a final coalgebra and therefore has an associated notion of behavioural equivalence. Then it is shown that behavioural equivalence of the functor coincides with $\mathcal{S}$-bisimilarity, 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. [37, 51]. Finally, it is worth noting that

for the coalgebraic treatment itself of *FuTS* we propose here it is not necessary for the co-domain of continuations to be semirings; working with monoids would be sufficient. However, the richer structure of semirings is convenient, if not essential, when using continuations and their operators in the formal definition of the *FuTS* semantics of *SPC*.

Using the bridge established by the *FuTS* bisimulation vs. coalgebraic behavioral equivalence correspondence results, we continue by showing for two well-known stochastic process algebras, viz. Hillston's *PEPA* [31] and Hermanns's *IML* [27], that the standard notions of *PEPA* strong equivalence and *IML* strong bisimilarity coincide with bisimilarity of the associated proper simple and combined *FuTS*, respectively. In turn, this means that the standard notions of strong equivalence and strong bisimilarity coincide with behavioural equivalence when cast in a coalgebraic framework.

PEPA stands out as one of the prominent Markovian process algebras, and *IML* specifically provides separate prefix constructions for actions and for delays. In passing, the issue of transition multiplicity has to be dealt with. Appropriate lemmas are provided relating the relation-based cumulative treatment with *FuTS* 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*. We furthermore provide a combined *FuTS* semantics for a simple language of deterministically-timed processes, viz. *TPC* [3] and we show the coincidence between *FuTS* bisimilarity and the standard equivalence of timed bisimilarity for the language. Finally, we provide a general *FuTS* semantics for a process language which incorporates non-determinism, discrete probabilities and Markovian randomized delays, i.e. a language for Markov Automata [22, 23]. Also in this case we prove that *FuTS* bisimulation and Markov Automata bisimulation coincide, adding to the claim that *FuTS* bisimulation is a natural notion of process identification for *SPC*.

Related work on coalgebra includes the papers [56, 35, 50]. Additionally, these papers cover measures and congruence formats, a topic not touched upon in the present paper. For what concerns the discrete parts, regarding the correspondence of bisimulations, our work is in line with the approach of the papers mentioned. In the treatment below the bi-algebraic perspective of SOS and bisimulation [55] is left implicit. In [41] an approach similar to ours has been applied to the ULTraS model, a model which shares some features with simple *FuTS*. In ULTraS posets are used instead of semirings, although a monoidal structure is then implicitly assumed when process equivalences are taken into consideration [7]. Furthermore, in [41] a general GSOS specification format is presented which allows for a 'syntactic' treatment of continuations involving so-called *weight functions*. 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 [11, 34, 49, 10]. Klin considers weighted transition systems, labelled transition systems that assign a weight to each transition and develops Weighted GSOS, a (meta-)syntactic framework for defining well-behaved weighted transition systems. For commutative monoids the notion of a weighted transition system compares with our notion of a *FuTS*, and, when cast in the coalgebraic setting, the associated concept of bisimulation coincides with behavioral equivalence. Weights of transitions of weighted transition systems are computed by induction on the syntax of process terms and by taking into account the contribution of all those GSOS rules that are triggered by the relevant (apparent) weights. Note that such a set of rules is finite. So, in a sense, the computation of the weights is *distributed among (the instantiations of) the relevant rules* with intermediate results collected (and integrated) in the final weight. In comparison, as mentioned before, in the *FuTS* approach, the relevant values are manipulated in a more direct way, using the higher-order operators on continuation functions, applying them directly to the continuations in the transitions *within the same* the semantics definition rules. So, in a

sense, the *FuTS* approach is better suited for a *compositional* definition of the operational semantics of a wide range of process calculi due to the suitable choice of a rich set of generic operations on continuation functions. In [39] the investigation on the relationship for nested *FuTS* between *FuTS* bisimilarity, and behavioural equivalence, and also coalgebraic bisimilarity is presented. In particular, it is shown that the functor type involved preserves weak pullbacks when the underlying semiring satisfies the zero-sum property.

The process languages with stochastic delays we consider in the sequel, 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. [33], since an explicit representation of such a distinction is not relevant for the subject of this paper. A coalgebraic treatment of this distinction is proposed in [57], 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 into the form $\mathcal{FS}(\cdot, \mathbb{B})^{\mathcal{L}}$ for $\mathcal{L} = \mathcal{P}(\mathcal{P}(\mathcal{A})) \times \mathcal{A}$, fitting the format of the functor for the (simple) *FuTS* considered here. In [12] processes are interpreted as formal power-series over a semiring in the style of [47]. This allows to compare testing equivalence for a CSP-style language and bisimulation in a Moore automaton. Note that the notions of equivalence addressed in this paper, as often in coalgebraic treatments of process relations, are all strong bisimilarities.

An extended abstract of part of this paper has appeared as [38] where the coalgebraic view of the *FuTS* approach and its application to *PEPA* and *IML* was originally presented. The workshop contribution [40] gives an account of bisimulation of *FuTS* of specific type and provides a general correspondence result with of *FuTS*-bisimulation and behavioral equivalence. The present paper covers these ideas in a structured way, going from simple *FuTS* to combined *FuTS* and nested *FuTS*. It includes the presentation of the use of combined *FuTS* for the definition of the semantics of a language of deterministically timed processes and the treatment of nested *FuTS* for the integration of stochastically timed, non-deterministic and probabilistic processes, as in Markov Automata.

For the present paper we assume the reader to have some familiarity with *SPC* and the application of *FuTS* for the definition of their semantics. The reader is referred to [18] for an introduction on the subject. Furthermore, in [39] an illustrative definition of a simple, qualitative, process calculus in the *FuTS* framework is shown. Section 2 provides basic concepts and notation. Simple *FuTS* are introduced in Section 3, followed by their coalgebraic treatment in Section 4. Simple *FuTS* are illustrated by the case of *PEPA* in Section 5 which also covers the correspondence of the respective notions of bisimulation. Section 6 introduces combined *FuTS* as well as their coalgebraic representation. Sections 7 and 8 treat *IML* and *TPC*. For both *SPC*, semantics based on combined *FuTS* are given, and *FuTS* bisimulation is compared to standard bisimulation. Next, Section 9 introduces nested as well as general *FuTS*, again tying up with behavior equivalence. In Section 10, a general *FuTS* is used for the semantics of a Markov Automata language, for which the notion of bisimulation is related to the standard one. Section 11 wraps up and discusses closing remarks.

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 $\mathbb{B} = \{\mathit{false}, \mathit{true}\}$ with disjunction as sum and conjunction as multiplication, the non-negative reals $\mathbb{R}_{\geqslant 0}$ with the standard operations,

and the powerset construct $\mathbf{2}^X$ for a set X with intersection and union as sum and multiplication, respectively. 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. We use the notation $\oplus\varphi$ to denote the value $\sum_{x \in X} \varphi(x)$ in $\mathcal{R}$.

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 $[\, x_1 \mapsto r_1, \ldots, x_n \mapsto r_n \,]$, or more compactly $[\, x_i \mapsto r_i]_{i=1}^n$, with $x_i \in X$ 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 $x \in X \setminus \{x_1, \ldots, x_n\}$. In particular $[\,]$, or more precisely $[\,]_{\mathcal{R}}$, is the constant function $x \mapsto 0$ and $\mathbf{D}_x^{\mathcal{R}} = [\, x \mapsto 1 \,]$ is the Dirac function on $\mathcal{R}$ for $x \in X$; in the sequel we will often drop the subscript or superscript $\mathcal{R}$ from $[\,]_{\mathcal{R}}$ and $\mathbf{D}_x^{\mathcal{R}}$, when the semiring $\mathcal{R}$ is clear from the context.

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 are φ and ψ. Slightly more generally, for functions $\varphi_i \in \mathcal{FS}(X, \mathcal{R})$ where $i = 1, \ldots, n$, we define the function $\sum_{i=1}^n \varphi_i$ in $\mathcal{FS}(X, \mathcal{R})$ by $(\sum_{i=1}^n \varphi_i)(x) = \sum_{i=1}^n \varphi_i(x)$. 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 $\mid$ guarantees that $\varphi \mid \psi$ is well-defined. Again, $\varphi \mid \psi$ is of finite support as are φ and ψ. Such an operation 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 $\mid$.

We recall some basic definitions from coalgebra. See e.g. [46] for more details. For a functor $\mathcal{F} : \mathbf{Set} \to \mathbf{Set}$ on the category **Set** of sets and functions, a coalgebra $\mathcal{X}$ of $\mathcal{F}$ is a set X together with a mapping $\alpha : X \to \mathcal{F}(X)$. A homomorphism between two $\mathcal{F}$-coalgebras $\mathcal{X} = (X, \alpha)$ and $\mathcal{Y} = (Y, \beta)$ is a function $f : X \to Y$ such that $\mathcal{F}(f) \circ \alpha = \beta \circ f$. An $\mathcal{F}$-coalgebra $(\Omega_{\mathcal{F}}, \omega_{\mathcal{F}})$ is called final or terminal, if there exists, for every $\mathcal{F}$-coalgebra $\mathcal{X} = (X, \alpha)$, a unique homomorphism $[\![\cdot]\!]_{\mathcal{F}}^{\mathcal{X}} : (X, \alpha) \to (\Omega_{\mathcal{F}}, \omega_{\mathcal{F}})$. Two elements x_1, x_2 of an $\mathcal{F}$-coalgebra $\mathcal{X}$ are called behavioural equivalent with respect to $\mathcal{F}$ if $[\![x_1]\!]_{\mathcal{F}}^{\mathcal{X}} = [\![x_2]\!]_{\mathcal{F}}^{\mathcal{X}}$, denoted $x_1 \approx_{\mathcal{F}}^{\mathcal{S}} x_2$. In the notation $[\![\cdot]\!]_{\mathcal{F}}^{\mathcal{X}}$ as well as $\approx_{\mathcal{F}}^{\mathcal{X}}$, the indication of the specific coalgebra $\mathcal{X}$ will be omitted when clear from the context.

A functor $\mathcal{F}$ is called accessible if it preserves κ-filtered colimits for some cardinal number κ. However, in the category **Set**, we have the following characterization of accessibility: for every set X and any element $\xi \in \mathcal{F}X$, there exists a subset $Y \subseteq X$ with $|Y| < \kappa$, such that $\xi \in \mathcal{F}Y$. It holds that a functor has a final coalgebra if it is κ-accessible for some cardinal number κ. See [2, 1].

A number of proofs of results on process languages $\mathcal{P}$ in this paper rely on so-called guarded induction [36]. 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 induction assumes the existence of a 'complexity' 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, in particular, $c(X) > c(P)$ if $X := P$. For all concrete process languages treated in this paper such a complexity function can be given straightforwardly. See [14] for more detail.

For convenience we collect here a number of abbreviations used in the sequel: *CTMC* and *DTMC* for the standard notions of Continuous-Time Markov Chains and Discrete-Time Markov Chains, respectively; *LTS* for Labelled Transition System, *RTS* for Rate Transition System, and *FuTS* for Labelled State-to-Function Transition System, the extension of *LTS* we focus on in this paper; *SPC* for Stochastic Process Calculus, referring to the class of process algebras featuring

a choice construct based on a non-negative exponential distribution; for specific process calculi and semantic models, viz. *PEPA* for Performance Evaluation Process Algebra, *IMC* for Interactive Markov Chains and *IML* for the *IMC*-based language used in this paper, *TPC* for an example Timed Process Calculus, *MA* for Markov Automata and *MAL* for the *MA*-based language used in this paper.

3. Simple State-to-Function Labelled Transition Systems

Below we introduce *simple FuTS*, i.e. *FuTS* with a single transition relation, which are sufficient for the definition of the semantics of many of the relevant stochastic process languages proposed in the literature (see [18] for details).

Definition 1. A *simple FuTS* $\mathcal{S}$, in full 'a simple state-to-function labelled transition system', over label set $\mathcal{L}$ and semiring $\mathcal{R}$, is a tuple $\mathcal{S} = (S, \rightarrowtail)$ where $\rightarrowtail \subseteq S \times \mathcal{L} \times \mathcal{FS}(S, \mathcal{R})$. •

In the sequel we omit the word 'simple' when this cannot give rise to confusion. Similar as for state-to-state transitions of *LTS*, for state-to-function transitions of *FuTS* we write $s \overset{\ell}{\rightarrowtail} \nu$ for $(s, \ell, \nu) \in \rightarrowtail$. Note that a (simple) *FuTS* over a label set $\mathcal{L}$ and a semiring $\mathcal{R}$ is reminiscent of a weighted automaton [21]. However, for a FuTS no output function is given, as is for a weighted automaton. To stress the relationship between *LTS* and *FuTS* we stick to the terminology and notion stemming from *LTS*.

For a *FuTS* $\mathcal{S} = (S, \rightarrowtail)$ the set S is called the set of states or the carrier set. We refer to $\rightarrowtail$ as the state-to-function transition relation of $\mathcal{S}$ or just as the transition relation. A *FuTS* $\mathcal{S}$ is called total and deterministic if, for all $s \in S$ and $\ell \in \mathcal{L}$, we have $s \overset{\ell}{\rightarrowtail} \nu$ for exactly one $\nu \in \mathcal{FS}(S, \mathcal{R})$. In such a situation, the state-to-function relation $\rightarrowtail$ corresponds to a function $S \to \mathcal{L} \to \mathcal{FS}(S, \mathcal{R})$. For the remainder of the paper, all *FuTS* we consider will be total and deterministic. It is noted that Definition 1 slightly differs in formulation from the one provided in [18].

As an example, Figure 1 displays a simple *FuTS* over the action set $\mathcal{A}$ and the semiring $\mathbb{R}_{\geqslant 0}$ of the non-negative real numbers with standard sum and multiplication. The functions ν_0 to ν_3 used in the example have the property that $\oplus \nu_i(s) = 1$, 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 $\mathbb{R}_{\geqslant 0}$, with its weights adding up to 1, is called a (reactive) probabilistic transition system [24].

Below it will be notationally convenient to consider a (total, deterministic and simple) *FuTS* as a tuple (S, θ) with transition function $\theta : S \to \mathcal{L} \to \mathcal{FS}(S, \mathcal{R})$, rather than using the form $(S, \rightarrowtail)$ that occurs more frequently for concrete examples in the literature. We will use the notation with transition functions $\theta : S \to \mathcal{L} \to \mathcal{FS}(S, \mathcal{R})$ to introduce the notion of bisimilarity for a simple *FuTS*.

Definition 2. Let $\mathcal{S} = (S, \theta)$ be a simple *FuTS* over label set $\mathcal{L}$ and semiring $\mathcal{R}$. An equivalence relation $R \subseteq S \times S$ is called an $\mathcal{S}$-bisimulation if $R(s_1, s_2)$ implies

$$\sum_{t' \in [t]_R} \theta(s_1)(\ell)(t') = \sum_{t' \in [t]_R} \theta(s_2)(\ell)(t') \tag{3.1}$$

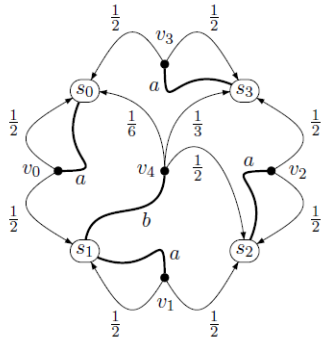

Figure 1: Simple *FuTS* for a probabilistic process.

for all $t \in S$ and $\ell \in \mathcal{L}$, 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 (3.1) exist since the functions $\theta(s_1)(\ell), \theta(s_2)(\ell) \in \mathcal{FS}(S, \mathcal{R})$ are of finite support.

4. Simple *FuTS* coalgebraically

In this section we will cast *simple FuTS* in the framework of coalgebras and prove a correspondence result of *FuTS* bisimilarity and behavioural equivalence for functors of the form $\mathcal{FS}(\cdot, \mathcal{R})^{\mathcal{L}}$ on **Set**, with $\mathcal{R}$ a semiring and $\mathcal{L}$ a set of labels.

Definition 3. Let $\mathcal{L}$ be a set of labels and $\mathcal{R}$ a semiring. Functor $\mathcal{U}^{\mathcal{L}}_{\mathcal{R}}$: **Set** $\to$ **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 $y \in Y$. •

Working in the context of *FuTS* we include the label set $\mathcal{L}$ in the notation for the functor $\mathcal{U}^{\mathcal{L}}_{\mathcal{R}}$. The functor $\mathcal{FS}(\cdot, \mathcal{X})$ itself, for $\mathcal{X}$ not necessarily a semiring, but a commutative monoid or field instead, have been studied frequently in the literature, see e.g. [25, 34, 10].

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. We 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 at comparing our notion of bisimilarity for simple *FuTS* with behavioural equivalence for the functor $\mathcal{U}^{\mathcal{L}}_{\mathcal{R}}$, $\mathcal{U}$ for short, 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, one may adapt the proof for the functor $\mathcal{FS}(\cdot, \mathcal{M})$: **Set** $\to$ **Set** where $\mathcal{M}$ is a monoid (rather than a semiring) as sketched in [48, 49] to the setting here. An alternative route to showing the existence of a final coalgebra is to verify accessibility. We directly apply the results of [2, Section 5].

Lemma 1. For a set of labels $\mathcal{L}$ and a semiring $\mathcal{R}$, the functor $\mathcal{U}$ has a final coalgebra.

Proof. It suffices to show that the **Set**-functor $\mathcal{U}$ is accessible for some suitable cardinal number. In fact, $\mathcal{U}$ is $|\mathcal{L}|\times\omega$-accessible: Consider $\varphi : \mathcal{L} \to \mathcal{FS}(X,\mathcal{R})$ in the image of the set X. Let $Y_\ell \subseteq X$ be the support of $\varphi(\ell) \in \mathcal{FS}(X,\mathcal{R})$ and $Y = \bigcup_{\ell\in\mathcal{L}} Y_\ell \subseteq X$. Then φ can be seen as an element of $\mathcal{L} \to \mathcal{FS}(Y,\mathcal{R})$, since outside of Y it holds that φ equals $0 \in \mathcal{R}$. □

Next we establish, for a given simple *FuTS* $\mathcal{S}$, the correspondence of $\mathcal{S}$-bisimulation as given by Definition 2 and behavioural equivalence induced by $\mathcal{U}$. The proof is similar to [10, Theorem 1].

Theorem 2. Let $\mathcal{S} = (S,\theta)$ be a simple *FuTS* over the label set $\mathcal{L}$ and semiring $\mathcal{R}$, and $\mathcal{U}$ as in Definition 3. Then $s_1 \simeq_{\mathcal{S}} s_2 \Leftrightarrow s_1 \approx_{\mathcal{U}} s_2$, for all $s_1, s_2 \in S$.

Proof. Let $s_1, s_2 \in S$. We first prove $s_1 \simeq_{\mathcal{S}} s_2 \Rightarrow s_1 \approx_{\mathcal{U}} s_2$. So, assume $s_1 \simeq_{\mathcal{S}} s_2$. Let $R \subseteq S \times S$ be an $\mathcal{S}$-bisimulation with $R(s_1, s_2)$. Note (S,θ) is a $\mathcal{U}$-coalgebra. We turn the collection of equivalence classes S/R into a $\mathcal{U}$-coalgebra $\mathcal{S}_R = (S/R, \varrho_R)$ where

$$\varrho_R([s]_R)(\ell)([t]_R) = \textstyle\sum_{t'\in[t]_R} \theta(s)(\ell)(t')$$

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

$$\begin{aligned}
&\mathcal{U}(\varepsilon_R)(\theta(s))(\ell)([t]_R)\\
&\quad= \textstyle\sum_{t'\in\varepsilon_R^{-1}([t]_R)} \theta(s)(\ell)(t') && \text{by definition of } \mathcal{U}\\
&\quad= \textstyle\sum_{t'\in[t]_R} \theta(s)(\ell)(t') && \text{by definition of } \varepsilon_R\\
&\quad= \varrho_R([s]_R)(\ell)([t]_R) && \text{by definition of } \varrho_R\\
&\quad= \varrho_R(\varepsilon_R(s))(\ell)([t]_R) && \text{by definition of } \varepsilon_R
\end{aligned}$$

Thus, $\mathcal{U}(\varepsilon_R)\circ\theta = \varrho\circ\varepsilon_R$, i.e. ε_R is a $\mathcal{U}$-homomorphism. Therefore, by uniqueness of a final morphism, we have $[\![\cdot]\!]^{\mathcal{S}}_{\mathcal{U}} = [\![\cdot]\!]^{\mathcal{S}_R}_{\mathcal{U}} \circ \varepsilon_R$. In particular, with respect to $\mathcal{S}$, this implies $[\![s_1]\!]_{\mathcal{U}} = [\![s_2]\!]_{\mathcal{U}}$ since $\varepsilon_R(s_1) = \varepsilon_R(s_2)$. Thus, $s_1 \approx_{\mathcal{U}} s_2$.

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

$$\omega([\![s]\!]_{\mathcal{U}})(\ell)(w) = \mathcal{U}([\![\cdot]\!]_{\mathcal{U}})(\theta(s))(\ell)(w) = \textstyle\sum_{t'\in[\![\cdot]\!]^{-1}_{\mathcal{U}}(w)} \theta(s)(\ell)(t') \tag{4.1}$$

Therefore we have

$$\begin{aligned}
&\textstyle\sum_{t'\in[t]_{R_{\mathcal{S}}}} \theta(s')(\ell)(t')\\
&\quad= \textstyle\sum_{t'\in[\![\cdot]\!]^{-1}_{\mathcal{U}}(w)} \theta(s')(\ell)(t') && \text{by definition of } R_{\mathcal{S}} \text{ and } w\\
&\quad= \omega([\![s']\!]_{\mathcal{U}})(\ell)(w) && \text{by equation (4.1)}\\
&\quad= \omega([\![s'']\!]_{\mathcal{U}})(\ell)(w) && s' \approx_{\mathcal{U}} s'' \text{ by assumption}\\
&\quad= \textstyle\sum_{t'\in[\![\cdot]\!]^{-1}_{\mathcal{U}}(w)} \theta(s'')(\ell)(t') && \text{by equation (4.1)}\\
&\quad= \textstyle\sum_{t'\in[t]_{R_{\mathcal{S}}}} \theta(s'')(\ell)(t') && \text{by definition of } R_{\mathcal{S}} \text{ and } w
\end{aligned}$$

Thus, if $R_{\mathcal{S}}(s',s'')$ then $\sum_{t'\in[t]_{R_{\mathcal{S}}}} \theta(s')(\ell)(t') = \sum_{t'\in[t]_{R_{\mathcal{S}}}} \theta(s'')(\ell)(t')$ for all $t \in S$ and $\ell \in \mathcal{L}$, and therefore $R_{\mathcal{S}}$ is an $\mathcal{S}$-bisimulation. Since $[\![s_1]\!]_{\mathcal{U}} = [\![s_2]\!]_{\mathcal{U}}$, 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 $s_1 \simeq_{\mathcal{S}} s_2$. □

In the next section we will provide *FuTS* semantics for a fragment of *PEPA*, a representative process language. For this language we will establish that its standard notion of strong equivalence as known in the literature coincides with the notion of strong bisimulation as induced by the *FuTS* semantics. The results of this section form the basis for showing that the standard notions of strong equivalence on the one hand, and behavioural equivalence 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 of *PEPA*

In this section we consider a significant fragment of the *Performance Evaluation Process Algebra*, *PEPA*, [31] –which we still call *PEPA* for simplicity– 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 yield a conceptual benefit for this paper. We present a *FuTS* semantics for *PEPA* in line with [18] and show that *PEPA*'s notion of equivalence $\sim_{pepa}$, called strong equivalence in [31], fits with the bisimilarity induced by the *FuTS* semantics.

Definition 4. The set $\mathcal{P}_{pepa}$ of *PEPA* processes is given by the grammar below:

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

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 $\mathcal{X}$. •

For $X \in \mathcal{X}$, the notation $X := P$ indicates that the process P is associated with the process 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 *SPC*, e.g. [29, 8], 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, synchronized execution of P and Q for the actions occurring in A, on the other hand. The transition rules of the *FuTS*-semantics of the fragment of *PEPA* we consider here is given in Figure 2, on which we comment below.

Characteristic for the *PEPA* language is the choice to model parallel composition, or cooperation in the terminology of *PEPA*, scaled by the minimum of the so-called apparent rates. By doing so, *PEPA*'s strong equivalence becomes a congruence [31]. Informally, 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 [31, 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 of the two processes, on average, 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

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

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

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

Figure 2: *FuTS* Transition Deduction System for *PEPA*.

execution yields the action a with rate $r_1 \cdot r_2 \cdot arf(P,Q)$. Here, the 'syntactic' scaling factor $arf(P,Q)$, the apparent rate factor, is defined by

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

assuming $r_a(P), r_a(Q) > 0$, otherwise $arf(P,Q) = 0$. Organizing the product $r_1 \cdot r_2 \cdot arf(P,Q)$ 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* we consider for the semantics of *PEPA* has been proposed originally in [18]. The transition relation is given by the rules in Figure 2. 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 the *FuTS* semantics considered in this paper, we conventionally use the special symbol δ for denoting that there is a random *delay*, with an negative exponential distribution, associated with the action. The underlying semiring for the *FuTS* for *PEPA* is the semiring $\mathbb{R}_{\geqslant 0}$ of non-negative reals.

Definition 5. The simple *FuTS* $\mathcal{S}_{pepa} = (\mathcal{P}_{pepa}, \rightarrowtail_{pepa})$ over $\Delta_{\mathcal{A}}$ and $\mathbb{R}_{\geqslant 0}$ has as transition relation the smallest relation satisfying the axioms and rules of Figure 2. •

We discuss the rules of $\mathcal{S}_{pepa}$. The *FuTS* semantics provides $\mathbf{nil} \overset{\delta_a}{\rightarrowtail}_{pepa} []_{\mathbb{R}_{\geqslant 0}}$, 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.

The semantics of the choice operator is defined by rule (CHO), where the continuation of process $P + Q$ is given by direct composition—using pointwise sum—of the continuation $\mathscr{P}$ of P and the continuation $\mathscr{Q}$ of Q.

Regarding the parallel operator $\underset{A}{\bowtie}$, with respect to some cooperation set $A \subseteq \mathcal{A}$ there are two rules. Now the distinction is between interleaving and synchronization. In the case of a label δ_a involving an action a not in the subset A, either the P-operand or the Q-operand of $P \underset{A}{\bowtie} Q$ 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

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

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

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

Figure 3: Standard Transition Deduction System for *PEPA*.

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 $P' \underset{A}{\bowtie} Q'$. Note that the syntactic constructor $\underset{A}{\bowtie} : \mathcal{P}_{pepa} \times \mathcal{P}_{pepa} \to \mathcal{P}_{pepa}$ is clearly injective; so, for all functions $\mathscr{P}$ and $\mathscr{Q}$ in $\mathcal{FS}(\mathcal{P}_{pepa}, \mathbb{R}_{\geqslant 0})$, we can define $\mathscr{P} \underset{A}{\bowtie} \mathscr{Q}$, as described in Section 2. 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 $arf(\mathscr{P}, \mathscr{Q})$ is applied to $\mathscr{P} \underset{A}{\bowtie} \mathscr{Q}$. This scaling factor for continuation in $\mathcal{FS}(\mathcal{P}_{pepa}, \mathbb{R}_{\geqslant 0})$, is, very much similar to its 'syntactic' counterpart, given by

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

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

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

Lemma 3. 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 [31]). It is also easy to prove, by guarded induction, that the *FuTS* $\mathcal{S}_{pepa}$ given by Definition 5 is total and deterministic.

Lemma 4. The *FuTS* $\mathcal{S}_{pepa}$ is total and deterministic. □

In view of the lemma it is justified to write $\mathcal{S}_{pepa} = (\mathcal{P}_{pepa}, \theta_{pepa})$. We use the abbreviated notation $\simeq_{pepa}$ for denoting $\simeq_{\mathcal{S}_{pepa}}$, the bisimulation equivalence induced by $\mathcal{S}_{pepa}$.

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* [31, 32] is given in Figure 3. 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 an appropriate

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. 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 [31, 32] for a subset of processes $C \subseteq \mathcal{P}_{pepa}$ and $a \in \mathcal{A}$ is given by

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

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 3. We are now ready to define *PEPA*'s notion of strong equivalence.

Definition 6. 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. The lemma is pivotal in relating *FuTS* bisimulation and standard bisimulation for *PEPA* in Theorem 6 below.

Lemma 5. 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 $\between_{\mathrm{A}} : \mathcal{P}_{pepa} \times \mathcal{P}_{pepa} \to \mathcal{P}_{pepa}$ with $\between_{\mathrm{A}}(P_1, P_2) = P_1 \between_{\mathrm{A}} P_2$ is injective. Recall, for $\mathscr{P}_1, \mathscr{P}_2 \in \mathcal{FS}(\mathcal{P}_{pepa}, \mathbb{R}_{\geqslant 0})$, we have $(\mathscr{P}_1 \between_{\mathrm{A}} \mathscr{P}_2)(P_1 \between_{\mathrm{A}} P_2) = \mathscr{P}_1(P_1) \cdot \mathscr{P}_2(P_2)$.

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

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

Case (II), $P' = P_1 \between_{\mathrm{A}} P'_2$, $P'_2 \neq P_2$: similar.

Case (III), $P' = P_1 \between_A P_2$. Then we have:

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

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

$$\begin{aligned}
&\textstyle\sum \{\!\{ \lambda \mid P_1 \between_A P_2 \xrightarrow{a,\lambda}_{pepa} P' \}\!\} \\
&\quad = \textstyle \sum \{\!\{ arf(P_1, P_2) \cdot \lambda_1 \cdot \lambda_2 \mid P_1 \xrightarrow{a,\lambda_1}_{pepa} P'_1,\ P_2 \xrightarrow{a,\lambda_2}_{pepa} P'_2 \}\!\} && \text{by rule (PAR2)} \\
&\quad = \textstyle arf(P_1, P_2) \cdot (\sum \{\!\{ \lambda_1 \mid P_1 \xrightarrow{a,\lambda_1}_{pepa} P'_1 \}\!\}) \cdot (\sum \{\!\{ \lambda_2 \mid P_2 \xrightarrow{a,\lambda_2}_{pepa} P'_2 \}\!\}) && \text{by distributivity} \\
&\quad = arf(P_1, P_2) \cdot \mathscr{P}_1(P'_1) \cdot \mathscr{P}_2(P'_2) && \text{by the induction hypothesis} \\
&\quad = arf(\mathscr{P}_1, \mathscr{P}_2) \cdot \mathscr{P}_1(P'_1) \cdot \mathscr{P}_2(P'_2) && \text{by Lemma 3} \\
&\quad = arf(\mathscr{P}_1, \mathscr{P}_2) \cdot (\mathscr{P}_1 \between_A \mathscr{P}_2)(P'_1 \between_A P'_2) && \text{definition } \between_A \text{ on } \mathcal{FS}(\mathcal{P}_{pepa}, \mathbb{R}_{\geqslant 0}) \\
&\quad = \mathscr{P}(P') && \text{by rule (PAR2)}
\end{aligned}$$

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 6.

Theorem 6. For *PEPA*-processes $P_1, P_2 \in \mathcal{P}_{pepa}$, it holds that $P_1 \simeq_{pepa} P_2$ 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

$$\begin{aligned}
q[P, [Q]_R, a] &= \textstyle \sum_{Q' \in [Q]_R} \sum \{\!\{ \lambda \mid P \xrightarrow{a,\lambda}_{pepa} Q' \}\!\} && \text{by definition } q[P, [Q]_R, a] \\
&= \textstyle \sum_{Q' \in [Q]_R} \mathscr{P}(Q') && \text{by Lemma 5} \\
&= \textstyle \sum_{Q' \in [Q]_R} \theta_{pepa}(P)(a)(Q') && \text{by definition } \theta_{pepa}
\end{aligned}$$

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 (Definition 6) iff R is an $\mathcal{S}_{pepa}$-bisimulation (Definition 2), from which the theorem follows. □

By the theorem the *FuTS* semantics for *PEPA* of Definition 5 is correct with respect to *PEPA*'s standard semantics of Figure 3. However, because of the use of continuation functions, the former does not involve implicit counting, decorations or multisets. From the general results on *FuTS* of the previous section, we also obtain a coalgebraic semantics for *PEPA* for which behavioral equivalence coincides with strong equivalence as defined in [31].

6. Combined *FuTS*

In the sequel of this article we will deal with a number of calculi and models that mix non-deterministic behaviour with stochastic or deterministic time or with probabilistic behaviour. In this section, we introduce the notion of a *combined FuTS*, which allows for a clean definition of the semantics of calculi where different aspects of behaviour are integrated in an orthogonal way. Prominent examples of such calculi are *IML*, a language for *IMC* where non-determinism is integrated with stochastic continuous delays (see Section 7) and *TPC*, a language where where non-determinism is integrated with deterministic discrete delays (see Section 8).

Definition 7. A combined *FuTS* $\mathcal{S}$, in full 'a combined state-to-function labeled transition system', over a number of label sets $\mathcal{L}_i$ and semirings $\mathcal{R}_i$, $i = 1, \ldots, n$, is a tuple $\mathcal{S} = (S, \langle \rightarrowtail_i \rangle_{i=1}^n)$ with set of states S and such that $\rightarrowtail_i \subseteq S \times \mathcal{L}_i \times \mathcal{FS}(S, \mathcal{R}_i)$, for $i = 1, \ldots, n$. •

Combined *FuTS* of Definition 7 extend the simple ones of Definition 1. Note, a combined *FuTS* is defined over a number of label sets and semirings, and, accordingly, gives rise to the same number of transition relations. Thus, a combined *FuTS* can be seen as a multi-dimensional simple *FuTS*. The underlying idea is that the behaviour model given by a combined *FuTS* is such that one can identify different *types* of labels, assuming disjoint label sets $\mathcal{L}_1, \ldots, \mathcal{L}_n$. Then, the continuation function of a transition labeled with an element of $\mathcal{L}_i$ is taken from $\mathcal{FS}(S, \mathcal{R}_i)$, expressing the association of the label set $\mathcal{L}_i$ with the semiring $\mathcal{R}_i$.

For example, in the case of *IML*, with set of processes $\mathcal{P}_{iml}$, both non-deterministic behaviour and stochastically-timed behaviour are treated. Furthermore, action execution is intended to be instantaneous, while stochastic time is characterized by the rates of negative exponential distributions. Consequently, it is convenient to use two label sets, namely a set of actions $\mathcal{A}$ and a singleton set $\Delta = \{\delta\}$ where the symbol δ is used as label to indicate that the transition involves some exponentially distributed delay. The relevant semirings will be $\mathbb{B}$, used for modeling the purely non-deterministic aspects of behaviour, and $\mathbb{R}_{\geqslant 0}$, used for the rates characterizing the stochastic aspects of behaviour, as in the case of *PEPA*, but here without any association of delay and actions. Consequently, for *IML* there will be two transition relations: $\rightarrowtail_1 \subseteq \mathcal{P}_{iml} \times \mathcal{A} \times \mathcal{FS}(\mathcal{P}_{iml}, \mathbb{B})$ modeling non-deterministic behaviour, and $\rightarrowtail_2 \subseteq \mathcal{P}_{iml} \times \Delta \times \mathcal{FS}(\mathcal{P}_{iml}, \mathbb{R}_{\geqslant 0})$ modeling stochastic-time behaviour.

It is worth pointing out here that one could use an alternative approach instead of taking resort to combined *FuTS*, namely one based on disjoint unions of label sets, and respectively, continuation functions. Letting $\bigoplus_{i=1}^n X_i$ denote the disjoint union of sets X_i, $i = 1, \ldots, n$, one could use a *single* transition relation

$$\rightarrowtail \subseteq S \times \bigoplus_{i=1}^{n} \mathcal{L}_i \times \bigoplus_{i=1}^{n} \mathcal{FS}(S, \mathcal{R}_i)$$

satisfying the additional property that $v \in \mathcal{FS}(S, \mathcal{R}_i)$ if $\ell \in \mathcal{L}_i$, for all transitions $s \overset{\ell}{\rightarrowtail} v$. As a matter of fact, this approach based on disjoint unions and a single transition relation has been used in [18]. Technically, the two approaches are equivalent. On the other hand, in the definition with a single transition relation, type compatibility between labels and continuation functions yields an additional proof obligation for the well-definedness the definition of the operational semantics for every specific process calculus (the interested reader is referred to [18] for details). The use of an approach with multiple transition relations instead, automatically guarantees type compatibility, viz. *by definition*. Furthermore, the approach based on disjoint unions appears less amenable to a category-theoretical treatment. For the reasons mentioned we stick to the format of Definition 7 in this paper.

As we will see, for the purposes of the present paper it is sufficient to consider only total and deterministic combined *FuTS*, i.e. those where every transition relation $\rightarrowtail_i$ is a total function. Consequently, it will be notationally convenient to consider a combined *FuTS* $\mathcal{S} = (S, \langle \rightarrowtail_i \rangle_{i=1}^n)$ as a tuple $(S, \langle \theta_i \rangle_{i=1}^n)$ with transition functions $\theta_i : S \to \mathcal{L}_i \to \mathcal{FS}(S, \mathcal{R}_i)$, for $i = 1, \ldots, n$, rather than using the form $(S, \langle \rightarrowtail_i \rangle_{i=1}^n)$ that occurs more frequently for concrete examples in the literature. In the sequel, we occasionally omit the qualification 'combined' for a combined *FuTS* when this cannot cause confusion. All relevant definitions and results presented in Sections 3 and 4 can be extended straightforwardly to combined *FuTS*. We refer to [39] for details on the extension of definitions, results and their proofs. Here we recall the most important ones.

Definition 8. For a combined *FuTS* $\mathcal{S} = (S, \langle \theta_i \rangle_{i=1}^n)$, an $\mathcal{S}$-bisimulation is an equivalence relation $R \subseteq S \times S$ such that $R(s_1, s_2)$ implies

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

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

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

Definition 9. Let $L = \langle \mathcal{L}_1, \ldots, \mathcal{L}_n \rangle$ be an n-tuple of label sets and $R = \langle \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$ be an n-tuple of semirings. The functor $\mathcal{V}_R^L$ on **Set** is defined by $\mathcal{V}_R^L = \prod_{i=1}^n \mathcal{FS}(\,\cdot\,, \mathcal{R}_i)^{\mathcal{L}_i}$.

Referring to Definition 3, we have $\mathcal{FS}(\,\cdot\,, \mathcal{R}_i)^{\mathcal{L}_i} = \mathcal{U}_{\mathcal{R}_i}^{\mathcal{L}_i}$, for $i = 1, \ldots, n$. Therefore, $\mathcal{V}_R^L = \prod_{i=1}^n \mathcal{U}_{\mathcal{R}_i}^{\mathcal{L}_i}$. We note that any combined *FuTS* $\mathcal{S} = (S, \langle \theta_i \rangle_{i=1}^n)$ over label sets $\mathcal{L}_i$ and semirings $\mathcal{R}_i$, for $i = 1, \ldots, n$, is in fact a $\mathcal{V}_R^L$-coalgebra. Reversely, every $\mathcal{V}_R^L$-coalgebra, for $L = \langle \mathcal{L}_1, \ldots, \mathcal{L}_n \rangle$ and $R = \langle \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$, corresponds to a combined *FuTS* over the label sets $\mathcal{L}_i$ and semirings $\mathcal{R}_i$, for $i = 1, \ldots, n$. Below we shall use $\mathcal{V}$ as an abbreviation for $\mathcal{V}_R^L$ whenever $L = \langle \mathcal{L}_1, \ldots, \mathcal{L}_n \rangle$ and $R = \langle \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$ are clear from the context. Similarly, for the sake of readability, we shall often abbreviate $\mathcal{U}_{\mathcal{R}_i}^{\mathcal{L}_i}$ by $\mathcal{U}_i$.

As product of accessible functors, the functor $\mathcal{V}$ of Definition 9 is accessible and possesses a final coalgebra, (Ω, ω) say. So, we can speak of the behavioural equivalence $\approx_{\mathcal{V}}$ on any $\mathcal{V}$-coalgebra or, equivalently, of any combined *FuTS* $\mathcal{S}$. Moreover, 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 $\simeq_{\mathcal{S}}$ and the behavioural equivalence $\approx_{\mathcal{V}}$ for the functor $\mathcal{V}$. Thus, one may argue, Definition 8 provides an explicit description of behavioral equivalence. The proof of the theorem below for combined *FuTS* is an adaptation of the proof of Theorem 2 for simple ones (see [39] for details).

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 9. Then $s_1 \simeq_{\mathcal{S}} s_2 \Leftrightarrow s_1 \approx_{\mathcal{V}} s_2$, for all $s_1, s_2 \in S$. □

In the sequel of the paper we will consider combined *FuTS*, as well as a so-called general *FuTS*, for concrete process languages. We will show for each process language that the notion of bisimulation of its *FuTS* coincides with the notion of strong bisimulation that is associated in the literature with the language. Consequently, as a corollary of Theorem 7, we obtain that the notions of strong bisimulations align with behavioral equivalence.

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 [27], *IMC* for short, and compare the notion of bisimulation induced by its *FuTS* to the standard notion of bisimulation based on the SOS-semantics as reported in the literature.

IMC 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* embody both non-deterministic behaviour and stochastic, i.e. stochastically timed, behaviour. System analysis using *IMC* proves to be a powerful approach because of the orthogonality of qualitative and quantitative dynamics, their logical underpinning and tool support, cf. [9, 30, 13]. Such orthogonality makes it natural to use a *combined FuTS* for the semantics of *IML*. A number of behavioural equivalences, both strong and weak, are available for *IMC* [22]. In our treatment here, we discuss a sublanguage of *IML*, which we still call *IML* for simplicity. In particular we do not deal with internal τ-steps, since we focus on strong bisimilarity here. The *FuTS* semantics we consider in the sequel has been originally proposed in [18].

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

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

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 (action) guardedness requirements for constant definitions and usage as in Section 5 for *PEPA*.

In line with the discussion above, in *IML* there are separate prefix constructions for actions $a.P$ (meaning that the process *instantaneously performs* action a and then behaves like P) and for time-delays $\lambda.P$ (meaning that the process is *delayed* for a period of time governed by a random variable with negative exponential distribution with rate λ, and then behaves like P). No restriction is imposed on the alternative and parallel composition of processes. For example, in *IML*, we have the process $a.\lambda.\mathbf{nil} + \mu.b.\mathbf{nil}$. With respect to the *FuTS* semantics to be defined below, we will see that this process admits both a non-trivial interactive transition and a non-trivial Markovian transition,

$$\begin{aligned} a.\lambda.\mathbf{nil} + \mu.b.\mathbf{nil} \ &\overset{a}{\rightarrowtail}_1 \ [\,\lambda.\mathbf{nil} \mapsto \mathit{true}\,] + [\,]_{\mathbb{B}} \ = \ [\,\lambda.\mathbf{nil} \mapsto \mathit{true}\,] \\ a.\lambda.\mathbf{nil} + \mu.b.\mathbf{nil} \ &\overset{\delta}{\rightarrowtail}_2 \ [\,]_{\mathbb{R}_{\geqslant 0}} + [\,b.\mathbf{nil} \mapsto \mu\,] \ = \ [\,b.\mathbf{nil} \mapsto \mu\,] \end{aligned}$$

leading to an interactive continuation and a Markovian continuation, respectively.

Definition 11. The *FuTS* semantics of $\mathcal{P}_{iml}$ is given by the *FuTS* $\mathcal{S}_{iml} = (\mathcal{P}_{iml}, \rightarrowtail_1, \rightarrowtail_2)$, a combined *FuTS* over the label sets $\mathcal{A}$ and $\Delta = \{\delta\}$ and the semirings $\mathbb{B}$ and $\mathbb{R}_{\geqslant 0}$ with transition relations $\rightarrowtail_1 \subseteq \mathcal{P}_{iml} \times \mathcal{A} \times \mathcal{FS}(\mathcal{P}_{iml}, \mathbb{B})$ and $\rightarrowtail_2 \subseteq \mathcal{P}_{iml} \times \Delta \times \mathcal{FS}(\mathcal{P}_{iml}, \mathbb{R}_{\geqslant 0})$ defined as the least relations satisfying the rules of Figure 4. •

$$\text{(NIL1)}\ \frac{a \in \mathcal{A}}{\mathbf{nil} \overset{a}{\rightarrowtail}_1 [\,]_{\mathbb{B}}} \qquad \text{(NIL2)}\ \frac{}{\mathbf{nil} \overset{\delta}{\rightarrowtail}_2 [\,]_{\mathbb{R}_{\geqslant 0}}} \qquad \text{(RPF1)}\ \frac{a \in \mathcal{A}}{\lambda.P \overset{a}{\rightarrowtail}_1 [\,]_{\mathbb{B}}}$$

$$\text{(APF1)}\ \frac{}{a.P \overset{a}{\rightarrowtail}_1 [P \mapsto \mathit{true}]} \qquad \text{(APF2)}\ \frac{b \neq a}{a.P \overset{b}{\rightarrowtail}_1 [\,]_{\mathbb{B}}} \qquad \text{(APF3)}\ \frac{}{a.P \overset{\delta}{\rightarrowtail}_2 [\,]_{\mathbb{R}_{\geqslant 0}}} \qquad \text{(RPF2)}\ \frac{}{\lambda.P \overset{\delta}{\rightarrowtail}_2 [P \mapsto \lambda]}$$

$$\text{(CHO1)}\ \frac{P \overset{a}{\rightarrowtail}_1 \mathscr{P} \quad Q \overset{a}{\rightarrowtail}_1 \mathscr{Q}}{P + Q \overset{a}{\rightarrowtail}_1 \mathscr{P} + \mathscr{Q}} \qquad \text{(CHO2)}\ \frac{P \overset{\delta}{\rightarrowtail}_2 \mathcal{P} \quad Q \overset{\delta}{\rightarrowtail}_2 \mathcal{Q}}{P + Q \overset{\delta}{\rightarrowtail}_2 \mathcal{P} + \mathcal{Q}}$$

$$\text{(PAR1)}\ \frac{P \overset{a}{\rightarrowtail}_1 \mathscr{P} \quad Q \overset{a}{\rightarrowtail}_1 \mathscr{Q} \quad a \notin A}{P \parallel_A Q \overset{a}{\rightarrowtail}_1 (\mathscr{P} \parallel_A \mathbf{D}_Q) + (\mathbf{D}_P \parallel_A \mathscr{Q})} \qquad \text{(PAR2)}\ \frac{P \overset{a}{\rightarrowtail}_1 \mathscr{P} \quad Q \overset{a}{\rightarrowtail}_1 \mathscr{Q} \quad a \in A}{P \parallel_A Q \overset{a}{\rightarrowtail}_1 \mathscr{P} \parallel_A \mathscr{Q}}$$

$$\text{(PAR3)}\ \frac{P \overset{\delta}{\rightarrowtail}_2 \mathcal{P} \quad Q \overset{\delta}{\rightarrowtail}_2 \mathcal{Q} \quad \delta \notin A}{P \parallel_A Q \overset{\delta}{\rightarrowtail}_2 (\mathcal{P} \parallel_A \mathbf{D}_Q) + (\mathbf{D}_P \parallel_A \mathcal{Q})} \qquad \text{(CON1)}\ \frac{P \overset{a}{\rightarrowtail}_1 \mathscr{P} \quad X := P}{X \overset{a}{\rightarrowtail}_1 \mathscr{P}} \qquad \text{(CON2)}\ \frac{P \overset{\delta}{\rightarrowtail}_2 \mathcal{P} \quad X := P}{X \overset{\delta}{\rightarrowtail}_2 \mathcal{P}}$$

Figure 4: *FuTS* Transition Deduction System for *IML*.

Actions $a \in \mathcal{A}$ decorate $\rightarrowtail_1$, the special symbol δ, with δ for delay, decorates $\rightarrowtail_2$. Note that 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 any action. In Figure 4 and in the rest of this section we use $\mathscr{P}, \mathscr{Q} \in \mathcal{FS}(\mathcal{P}_{iml}, \mathbb{B})$ as typical interactive continuations, and $\mathcal{P}, \mathcal{Q} \in \mathcal{FS}(\mathcal{P}_{iml}, \mathbb{R}_{\geqslant 0})$ as typical Markovian continuations.

For the parallel construct $\parallel_A$, interleaving applies both for non-synchronized actions $a \notin A$ as well as for delays. Therefore we have rule (PAR1) pertaining to $\rightarrowtail_1$ and rule (PAR3) pertaining to $\rightarrowtail_2$. The same holds for non-deterministic choice, rules (CHO1) and (CHO2), and constants, rules (CON1) and (CON2). Finally, *IML* does not provide synchronization of delays in the parallel construct. Hence, rule (PAR2) only concerns the transition relation $\rightarrowtail_1$ capturing synchronization on actions. We recall that for all $R \in \mathcal{P}_{iml}$, on the one hand,

$$(\mathscr{P} \parallel_A \mathscr{Q})(R) \;=\; \begin{cases} \mathscr{P}(R_1) \wedge \mathscr{Q}(R_2) & \text{if } R = R_1 \parallel_A R_2 \text{ for some } R_1, R_2 \in \mathcal{P}_{iml} \\ \mathit{false} & \text{otherwise} \end{cases}$$

and, on the other hand,

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

where $\cdot$ is the product in $\mathbb{R}_{\geqslant 0}$.

Example For $a.(\lambda.\mathbf{nil} + b.\mathbf{nil})$, $\mu.a.\mathbf{nil} \in \mathcal{P}_{iml}$ and $A = \{a\}$ we have

$$\begin{aligned} & a.(\lambda.\mathbf{nil} + b.\mathbf{nil}) \parallel_A \mu.a.\mathbf{nil} \\ & \quad \overset{\delta}{\rightarrowtail}_2 \quad [\,]_{\mathbb{R}_{\geqslant 0}} \parallel_A \mathbf{D}_{\mu.a.\mathbf{nil}} \;+\; \mathbf{D}_{a.(\lambda.\mathbf{nil}+b.\mathbf{nil})} \parallel_A [\, a.\mathbf{nil} \mapsto \mu \,] \\ & \quad = \quad [\,]_{\mathbb{R}_{\geqslant 0}} \parallel_A [\, \mu.a.\mathbf{nil} \mapsto 1 \,] \;+\; [a.(\lambda.\mathbf{nil} + b.\mathbf{nil}) \mapsto 1 \,] \parallel_A [\, a.\mathbf{nil} \mapsto \mu \,] \\ & \quad = \quad [\, a.(\lambda.\mathbf{nil} + b.\mathbf{nil}) \parallel_A a.\mathbf{nil} \;\mapsto\; \mu \,] \end{aligned}$$

For $X := a.\lambda.b.X$ and $Y := a.\mu.b.Y$, and $A = \{a, b\}$ we have

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

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

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

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

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

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

Figure 5: Standard Transition Deduction System for *IML*.

It is not difficult to verify that $\mathcal{S}_{iml}$ is a total and deterministic combined *FuTS*.

Lemma 8. The *FuTS* $\mathcal{S}_{iml}$ is total and deterministic. □

Below we use $\mathcal{S}_{iml} = (\mathcal{P}_{iml}, \theta_1, \theta_2)$ and write $\simeq_{iml}$ rather than $\simeq_{\mathcal{S}_{iml}}$, the bisimulation equivalence induced by $\mathcal{S}_{iml}$.

The standard SOS semantics of *IML* [27] is given in Figure 5 involving the transition relations

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

Below we will use functions $\mathbf{T}$ and $\mathbf{R}$ based on $\longrightarrow$ and $\dashrightarrow$, cf. [30]. We have $\mathbf{T}\colon \mathcal{P}_{iml} \times \mathcal{A} \times \mathbf{2}^{\mathcal{P}_{iml}} \to \mathbb{B}$ given by $\mathbf{T}(P, a, C) = \mathit{true}$ 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 $\mathbf{R}(P, P') = \sum \{\!|\, \lambda \mid P \xrightarrow{\lambda} P' \,|\!\}$. 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, $\mathit{cnt} : \mathcal{P}_{iml} \times \mathbb{R}_{>0} \times \mathcal{P}_{iml} \to \mathbb{R}_{\geqslant 0}$ returning the number of multiplicities of a transition $P \xrightarrow{\lambda} P'$, or other means of decorations. We extend $\mathbf{R}$ to $\mathcal{P}_{iml} \times \mathbf{2}^{\mathcal{P}_{iml}}$ by $\mathbf{R}(P, C) = \sum_{P' \in C} \sum \{\!|\, \lambda \mid P \xrightarrow{\lambda} P' \,|\!\}$, for $P \in \mathcal{P}_{iml}$, $C \subseteq \mathcal{P}_{iml}$.

For *IML* we have the following notion of strong bisimulation [27, 30] 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 $Q \in \mathcal{P}_{iml}$: $\mathbf{T}(P_1, a, [Q]_R) \iff \mathbf{T}(P_2, a, [Q]_R)$
- for all $Q \in \mathcal{P}_{iml}$: $\mathbf{R}(P_1, [Q]_R) = \mathbf{R}(P_2, [Q]_R)$.

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 $\simeq_{iml}$ for $\mathcal{S}_{iml}$ as given by Definition 11 and strong bisimilarity $\sim_{iml}$ for *IML* as given by Definition 12, we need to connect the state-to-function relation $\rightarrowtail_1$ and the transition relation $\longrightarrow$ as well as the state-to-function relation $\rightarrowtail_2$ and the transition relation $\dashrightarrow$.

Lemma 9.

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

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

Proof. (a) Guarded induction. Let $a \in \mathcal{A}$. We treat two typical cases, viz. $\lambda.P$ 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 $\mathscr{P} = [\,]_{\mathbb{B}}$. We have $\lambda.P \xrightarrow{a} P'$ for no $P' \in \mathcal{P}_{iml}$, as no transition is provided in $\rightarrow$, and we have $\mathscr{P}(P') = \mathit{false}$ 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 $Q \in \mathcal{P}_{iml}$, by definition of $\mathbf{D}_Q \in \mathcal{FS}(\mathcal{P}_{iml}, \mathbb{B})$, $\mathbf{D}_Q(Q') = \mathit{true}$ iff $Q' = Q$, for $Q' \in \mathcal{P}_{iml}$. We have

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

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 $\mu.P \overset{\delta}{\rightarrowtail}_2 \mathcal{P}$, then $\mathcal{P} = [\, P \mapsto \mu \,]$. Moreover, it holds that $\mu.P$ admits a single $\dashrightarrow$-transition, viz. $\mu.P \overset{\mu}{\dashrightarrow} P$. Thus we have $\sum \{\!|\, \lambda \mid \mu.P \overset{\lambda}{\dashrightarrow} P' \,|\!\} = \mu = [\, P \mapsto \mu \,](P) = \mathcal{P}(P)$.

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

$$
\begin{aligned}
& \sum \{\!|\, \lambda \mid P_1 \parallel_A P_2 \overset{\lambda}{\dashrightarrow} P' \,|\!\} \\
& \quad = \sum \{\!|\, \lambda \mid P_1 \overset{\lambda}{\dashrightarrow} P_1',\ P' = P_1' \parallel_A P_2 \,|\!\} + \sum \{\!|\, \lambda \mid P_2 \overset{\lambda}{\dashrightarrow} P_2',\ P' = P_1 \parallel_A P_2' \,|\!\} \\
& \qquad \text{by analysis of } \dashrightarrow \\
& \quad = (\,\texttt{if}\ P' = P_1' \parallel_A P_2\ \texttt{then} \sum \{\!|\, \lambda \mid P_1 \overset{\lambda}{\dashrightarrow} P_1' \,|\!\}\ \texttt{else}\ 0\ \texttt{end}\,) + \\
& \qquad\qquad (\,\texttt{if}\ P' = P_1 \parallel_A P_2'\ \texttt{then} \sum \{\!|\, \lambda \mid P_2 \overset{\lambda}{\dashrightarrow} P_2' \,|\!\}\ \texttt{else}\ 0\ \texttt{end}\,)
\end{aligned}
$$

$$
\begin{aligned}
&= (\,\texttt{if}\ P' = P_1' \parallel_A P_2\ \texttt{then}\ \mathcal{P}_1(P_1')\ \texttt{else}\ 0\ \texttt{end}\,) + \\
&\qquad (\,\texttt{if}\ P' = P_1 \parallel_A P_2'\ \texttt{then}\ \mathcal{P}_2(P_2')\ \texttt{else}\ 0\ \texttt{end}\,) \\
&\quad \text{by induction hypothesis for } P_1 \text{ and } P_2 \\
&= (\mathcal{P}_1 \parallel_A \mathbf{D}_{P_2})(P') + (\mathbf{D}_{P_1} \parallel_A \mathcal{P}_2)(P') \\
&\quad \text{by definition of } \parallel_A, \mathbf{D}_{P_1}, \mathbf{D}_{P_2} \text{ and } + \text{ on } \mathcal{FS}(\mathcal{P}_{iml}, \mathbb{R}_{\geqslant 0}) \\
&= \mathcal{P}(P')
\end{aligned}
$$

The remaining cases are left to the reader. □

We are now in a position to relate *FuTS* bisimilarity and standard strong bisimilarity for *IML*. In essence, Lemma 9 is all we need.

Theorem 10. For any two processes $P_1, P_2 \in \mathcal{P}_{iml}$ it holds that $P_1 \simeq_{iml} P_2$ 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 $P \overset{a}{\rightarrowtail}_1 \mathscr{P}$. Thus $\theta_1(P)(a) = \mathscr{P}$. Then we have

$$
\begin{aligned}
\mathbf{T}(P, a, [Q]_R) &\Leftrightarrow \exists Q' \in [Q]_R \colon P \xrightarrow{a} Q' && \text{by definition of } \mathbf{T} \\
&\Leftrightarrow \exists Q' \in [Q]_R \colon \mathscr{P}(Q') = \mathit{true} && \text{by Lemma 9a} \\
&\Leftrightarrow \textstyle\sum_{Q' \in [Q]_R} \theta_1(P)(a)(Q') = \mathit{true} && \text{by definition of } \theta_1
\end{aligned}
$$

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

$$
\begin{aligned}
\mathbf{R}(P, [Q]_R) &= \textstyle\sum_{Q' \in [Q]_R} \sum \{\!|\, \lambda \mid P \overset{\lambda}{\dashrightarrow} Q' \,|\!\} && \text{by definition of } \mathbf{R} \\
&= \textstyle\sum_{Q' \in [Q]_R} \mathcal{P}(Q') && \text{by Lemma 9b} \\
&= \textstyle\sum_{Q' \in [Q]_R} \theta_2(P)(\delta)(Q) && \text{by definition of } \theta_2
\end{aligned}
$$

We conclude that a strong bisimulation for *IML* is also an $\mathcal{S}_{iml}$-bisimulation for the *pFuTS* $\mathcal{S}_{iml}$, and vice versa. From this the theorem follows. □

From the theorem we conclude that also for *IML* the concrete notion of strong bisimilarity $\sim_{iml}$ is coalgebraically underpinned, as it coincides with the behavioral equivalence $\simeq_{iml}$ that comes with the corresponding *FuTS* $\mathcal{S}_{iml}$.

8. *FuTS* Semantics of *TPC*

In this section we consider a simple language of timed processes for which we provide a combined *FuTS*. The language is a relevant fragment of the timed process algebra *TPC* presented in [3]. The model of time under consideration is discrete and deterministic. The relevant construct is the time-prefix $(n).P$, with $n \in \mathbb{N}$, $n > 0$, expressing that the process P is to be executed after n time steps. We will provide a *FuTS* semantics and compare the induced notion of bisimulation to the notion of timed bisimulation underlying the operational semantics reported in [3].

To the best of our knowledge, this is the first time a deterministically timed model is dealt with in the coalgebraic framework. As we will see, we resort to $\mathbf{2}^{\mathbb{N}}$ as co-domain for the time continuations, instead of just $\mathbb{N}$, as one may expect. In particular, we use the semiring $\mathbf{2}^{\mathbb{N}}$ with set union as sum and intersection as multiplication. The reason of this choice is mainly technical and is connected to the proof of the bisimulation correspondence theorem (Theorem 14 below). Furthermore, the appropriate treatment of delays requires the extension of the set of operators on continuations.

$$\text{(NIL1)}\ \frac{a \in \mathcal{A}}{\mathbf{nil} \overset{a}{\rightarrowtail}_1 [\,]_{\mathbb{B}}} \qquad \text{(NIL2)}\ \frac{}{\mathbf{nil} \overset{\surd}{\rightarrowtail}_2 [\,]_{\mathbf{2}^{\mathbb{N}}}}$$

$$\text{(APF1)}\ \frac{}{a.P \overset{a}{\rightarrowtail}_1 [P \mapsto \mathit{true}]} \qquad \text{(APF2)}\ \frac{b \neq a}{a.P \overset{b}{\rightarrowtail}_1 [\,]_{\mathbb{B}}} \qquad \text{(APF3)}\ \frac{}{a.P \overset{\surd}{\rightarrowtail}_2 [\,]_{\mathbf{2}^{\mathbb{N}}}}$$

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

$$\text{(CHO1)}\ \frac{P \overset{a}{\rightarrowtail}_1 \mathscr{P} \quad Q \overset{a}{\rightarrowtail}_1 \mathscr{Q}}{P + Q \overset{a}{\rightarrowtail}_1 \mathscr{P} + \mathscr{Q}} \qquad \text{(CHO2)}\ \frac{P \overset{\surd}{\rightarrowtail}_2 \mathcal{P} \quad Q \overset{\surd}{\rightarrowtail}_2 \mathcal{Q}}{P + Q \overset{\surd}{\rightarrowtail}_2 \mathcal{P}\ [+]\ \mathcal{Q}}$$

$$\text{(PAR1)}\ \frac{P \overset{a}{\rightarrowtail}_1 \mathscr{P} \quad Q \overset{a}{\rightarrowtail}_1 \mathscr{Q} \quad a \notin A}{P \parallel_A Q \overset{a}{\rightarrowtail}_1 (\mathscr{P} \parallel_A \mathbf{D}_Q) + (\mathbf{D}_P \parallel_A \mathscr{Q})} \qquad \text{(PAR2)}\ \frac{P \overset{a}{\rightarrowtail}_1 \mathscr{P} \quad Q \overset{a}{\rightarrowtail}_1 \mathscr{Q} \quad a \in A}{P \parallel_A Q \overset{a}{\rightarrowtail}_2 \mathscr{P} \parallel_A \mathscr{Q}}$$

$$\text{(PAR3)}\ \frac{P \overset{\surd}{\rightarrowtail}_2 \mathcal{P} \quad Q \overset{\surd}{\rightarrowtail}_2 \mathcal{Q}}{P \parallel_A Q \overset{\surd}{\rightarrowtail}_2 \mathcal{P}\ [\parallel_A]\ \mathcal{Q}} \qquad \text{(CON1)}\ \frac{P \overset{\alpha}{\rightarrowtail}_1 \mathscr{P} \quad X := P}{X \overset{\alpha}{\rightarrowtail}_1 \mathscr{P}} \qquad \text{(CON2)}\ \frac{P \overset{\alpha}{\rightarrowtail}_2 \mathcal{P} \quad X := P}{X \overset{\alpha}{\rightarrowtail}_2 \mathcal{P}}$$

Figure 6: *FuTS* Transition Deduction System for *TPC*.

Definition 13. The set $\mathcal{P}_{tpc}$ of *TPC* processes is given by the grammar below:

$$P \ ::= \ \mathbf{nil} \mid a.P \mid (n)\,.\,P \mid P + P \mid P \parallel_A P \mid X$$

where a ranges over the set of actions $\mathcal{A}$, n over $\mathbb{N}$ with $n > 0$, 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 for *PEPA* or *IML*.

Definition 14. The formal semantics of $\mathcal{P}_{tpc}$ is given by the *FuTS* $\mathcal{S}_{tpc} = (\mathcal{P}_{tpc}, \rightarrowtail_1, \rightarrowtail_2)$, a combined *FuTS* over the label sets $\mathcal{A}$ and Θ with $\Theta = \{\surd\}$ and the semirings $\mathbb{B}$ and $\mathbf{2}^{\mathbb{N}}$ with transition relations $\rightarrowtail_1 \subseteq \mathcal{P}_{tpc} \times \mathcal{A} \times \mathcal{FS}(\mathcal{P}_{tpc}, \mathbb{B})$ and $\rightarrowtail_2 \subseteq \mathcal{P}_{tpc} \times \Theta \times \mathcal{FS}(\mathcal{P}_{tpc}, \mathbf{2}^{\mathbb{N}})$ defined as the least relations satisfying the rules of Figure 6. •

Also $\mathcal{S}_{tpc}$ is a *combined FuTS*, having the two state-to-function relations $\rightarrowtail_1$ and $\rightarrowtail_2$. Actions $a \in \mathcal{A}$ decorate $\rightarrowtail_1$, the special symbol $\surd$ decorates $\rightarrowtail_2$ (with a similar role as δ for *IML*). As for $\rightarrowtail_2$ the label is always the same, we occasionally suppress it. Note rule (APF3) and rule (TPF1) involve the null-functions of $\mathbf{2}^{\mathbb{N}}$ and of $\mathbb{B}$, respectively, to express that a process $a.P$ does not trigger a delay and a process $(n).P$ does not execute an action. In Figure 6 and in the rest of this section we use $\mathscr{P}, \mathscr{Q} \in \mathcal{FS}(\mathcal{P}_{tpc}, \mathbb{B})$ as typical action continuations, and $\mathcal{P}, \mathcal{Q} \in \mathcal{FS}(\mathcal{P}_{tpc}, \mathbf{2}^{\mathbb{N}})$ as typical time continuations.

The second time prefix rule (TPF2) combines a possible evolution over time of the process P into its continuation $\mathcal{P}$ with the elapse of the prefix. Note, the continuation in the conclusion of rule (TPF2) is a sum of three parts, viz. $[n;P]$, $[\,P \mapsto \{n\}\,]$, and $(n + \mathcal{P})$. The auxiliary mappings $[n;P]$ and $(n + \mathcal{P})$, for timed continuations, are given by

$$[n;P](Q) = \begin{cases} \{m\} & \text{if } Q = (n-m).P,\ 0 < m < n \\ \varnothing & \text{otherwise} \end{cases} \qquad (n + \mathcal{P})(Q) = \{\, n + m \mid m \in \mathcal{P}(Q) \,\}$$

It is easy to see that, for $n \in \mathbb{N}$, $Q \in \mathcal{P}_{tpc}$, and $\mathcal{P} \in \mathcal{FS}(\mathcal{P}_{tpc}, \mathbf{2}^{\mathbb{N}})$, $[n; Q] = [\,(n-i).Q \mapsto \{i\}\,]_{i=1}^{n-1}$, and if $\mathcal{P}(Q) = \emptyset$, then also $(n + \mathcal{P})(Q) = \emptyset$. 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, i.e. the component $[P \mapsto \{n\}]$ is used). After more than n time steps, say $n + m$ time steps, process Q is to be executed if $m \in \mathcal{P}(Q)$. Thus, if no such m exist, i.e. if $\mathcal{P}(Q) = \emptyset$, this yields an empty set too.

The rules for the choice and parallel construct of *TPC* make use of corresponding operations on $\mathcal{FS}(\mathcal{P}_{tpc}, \mathbb{B})$ and $\mathcal{FS}(\mathcal{P}_{tpc}, \mathbf{2}^{\mathbb{N}})$. For $\mathscr{P}, \mathscr{Q} \in \mathcal{FS}(\mathcal{P}_{tpc}, \mathbb{B})$, the functions $\mathscr{P} + \mathscr{Q}$ and $\mathscr{P} \parallel_A \mathscr{Q}$ are as before. For $\mathcal{FS}(\mathcal{P}_{tpc}, \mathbf{2}^{\mathbb{N}})$ the following operators are used:

$$(\mathcal{P}\ [+]\ \mathcal{Q})(R) = \begin{cases} \mathcal{P}(P) \cap \mathcal{Q}(Q) & \text{if } R = P + Q \text{ for } P, Q \in \mathcal{P}_{tpc} \\ \emptyset & \text{otherwise} \end{cases}$$

and, likewise

$$(\mathcal{P}\ [\,\|_A\,]\ \mathcal{Q})(R) = \begin{cases} \mathcal{P}(P) \cap \mathcal{Q}(Q) & \text{if } R = P \parallel_A Q, \text{ for } P, Q \in \mathcal{P}_{tpc} \\ \emptyset & \text{otherwise} \end{cases}$$

We have that for $P \in \mathcal{P}_{tpc}$ there exists a unique $\mathcal{P} \in \mathcal{FS}(\mathcal{P}_{tpc}, \mathbf{2}^{\mathbb{N}})$ such that $P \rightarrowtail_2 \mathcal{P}$. Moreover, given the rules for $\mathcal{S}_{tpc}$ and the definition of the operators above, it can verified that, for $P, Q \in \mathcal{P}_{tpc}$ and $\mathcal{P} \in \mathcal{FS}(\mathcal{P}_{tpc}, \mathbf{2}^{\mathbb{N}})$ such that $P \rightarrowtail_2 \mathcal{P}$ it holds that $\mathcal{P}(Q)$ is either a singleton or the empty set. See Lemma 11 below.

In order to prove the lemma we introduce an auxiliary function $md : \mathcal{P}_{tpc} \to \mathbb{N}$, establishing the so-called maximum delay of a process, given by

$$\begin{array}{rclcrcl} md(\mathbf{nil}) & = & 0 & & md(P_1 + P_2) & = & \min\{\, md(P_1),\, md(P_2)\,\} \\ md(a.P) & = & 0 & & md(P_1 \parallel_A P_2) & = & \min\{\, md(P_1),\, md(P_2)\,\} \\ md((n).P) & = & n + md(P) & & md((X) & = & md(P) \quad \text{if } X := P \end{array}$$

By guarded induction, one straightforwardly verifies the property that $md(Q') < md(Q)$ for $Q, Q' \in \mathcal{P}_{tpc}$ and $\mathcal{Q} \in \mathcal{FS}(\mathcal{P}_{tpc}, \mathbb{N})$ such that $Q \rightarrowtail_2 \mathcal{Q}$ and $\mathcal{Q}(Q') \neq \emptyset$. From this observation is follows that $[n; P]$, $[P \mapsto \{n\}]$ and $(n + \mathcal{P})$ have disjoint supports: We have that (i) if $[n; P](P') \neq \emptyset$ then $P' = (n-m).P$ for $0 < m < n$, hence $md(P') = (m-n) + md(P) > md(P)$; (ii) if $[\,P \mapsto \{n\}\,](P') \neq \emptyset$ then $P' = P$, hence $md(P') = md(P)$; (iii) if $(n + \mathcal{P})(P') \neq \emptyset$ then $\mathcal{P}(P') \neq \emptyset$ hence, using the property above, $md(P') < md(P)$.

Lemma 11.

(a) The *FuTS* $\mathcal{S}_{tpc}$ is total and deterministic.

(b) If $P \overset{\surd}{\rightarrowtail}_2 \mathcal{P}$ then either $\mathcal{P}(Q) = \{n\}$ for some $n > 0$ or $\mathcal{P}(Q) = \emptyset$.

Proof. Part (a) goes by guarded induction on P, both for $\rightarrowtail_1$ and $\rightarrowtail_2$. Part (b) follows by guarded induction. For the time prefix $(n).P$ we use that $[n; P]$, $[P \mapsto \{n\}]$ and $(n + \mathcal{P})$ have disjoint supports, as noted above. For the constructs $P + Q$ and $P \parallel_A Q$ we observe that the operations $[+]$ and $[\,\|_A\,]$ preserve the property mentioned, as the intersection of two singletons holding a positive number is either a singleton with a positive number or the empty set. □

Below we have $\mathcal{S}_{tpc} = (\mathcal{P}_{tpc}, \theta_1, \theta_2)$ and use $\simeq_{tpc}$ to denote the bisimulation equivalence induced by $\mathcal{S}_{tpc}$.

The standard SOS semantics of the *TPC* fragment of interest is given in Figure 7, involving the transition relations

$$\rightarrow \subseteq \mathcal{P}_{tpc} \times \mathcal{A} \times \mathcal{P}_{tpc} \qquad \text{and} \qquad \rightsquigarrow \subseteq \mathcal{P}_{tpc} \times \mathbb{N}_{>0} \times \mathcal{P}_{tpc}$$

$$\text{(APF)}\ \frac{}{a.P \xrightarrow{a} P} \qquad \text{(PRE)}\ \frac{}{(n).P \overset{n}{\rightsquigarrow} P} \qquad \text{(DEC)}\ \frac{n = m + \ell}{(n).P \overset{m}{\rightsquigarrow} (\ell).P} \qquad \text{(SUM)}\ \frac{P \overset{n}{\rightsquigarrow} P'}{(m).P \overset{n+m}{\rightsquigarrow} P'}$$

$$\text{(CHO1)}\ \frac{P \xrightarrow{a} R}{P + Q \xrightarrow{a} R} \qquad \text{(CHO2)}\ \frac{Q \xrightarrow{a} R}{P + Q \xrightarrow{a} R} \qquad \text{(ALT)}\ \frac{P \overset{n}{\rightsquigarrow} P' \quad Q \overset{n}{\rightsquigarrow} Q'}{P + Q \overset{n}{\rightsquigarrow} P' + Q'}$$

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

$$\text{(SYN)}\ \frac{P \overset{n}{\rightsquigarrow} P' \quad Q \overset{n}{\rightsquigarrow} Q'}{P \parallel_A Q \overset{n}{\rightsquigarrow} P' \parallel_A Q'} \qquad \text{(CON1)}\ \frac{P \xrightarrow{a} Q \quad X := P}{X \xrightarrow{a} Q} \qquad \text{(CON2)}\ \frac{P \overset{n}{\rightsquigarrow} Q \quad X := P}{X \overset{n}{\rightsquigarrow} Q}$$

Figure 7: Standard Transition Deduction System for *TPC*.

Note that for timed transitions $P \overset{n}{\rightsquigarrow} P'$ it is required that $n > 0$. Therefore, regarding rule (DEC), a process $(n).P$ for example with a timed prefix will *not* yield a zero-time step $(n).P \overset{0}{\rightsquigarrow} (n).P$ for which time does not progress. The case for $(n).P$ where n time step elapse, is covered by rule (PRE).

The definition of timed bisimilarity for *TPC* we give below is a bit more concise than the one originally introduced in [3], but the two notions can be easily proven to coincide. We will compare timed bisimilarity with the notion of bisimulation associated with the combined *FuTS* $\mathcal{S}_{tpc}$.

Definition 15. An equivalence relation $R \subseteq \mathcal{P}_{tpc} \times \mathcal{P}_{tpc}$ is a timed bisimulation for *TPC* if, for all $P_1, P_2 \in \mathcal{P}_{tpc}$ such that $R(P_1, P_2)$, it holds that for all $a \in \mathcal{A}$ and $n \in \mathbb{N}$

- whenever $P_1 \xrightarrow{a} Q_1$, then $P_2 \xrightarrow{a} Q_2$ for some $Q_2 \in \mathcal{P}_{tpc}$ with $R(Q_1, Q_2)$;
- whenever $P_1 \overset{n}{\rightsquigarrow} Q_1$, then $P_2 \overset{n}{\rightsquigarrow} Q_2$ for some $Q_2 \in \mathcal{P}_{tpc}$ with $R(Q_1, Q_2)$.

Two processes $P_1, P_2 \in \mathcal{P}_{tpc}$ are called timed bisimilar, notation $P_1 \sim_{tpc} P_2$ if $R(P_1, P_2)$ for some timed bisimulation for $\mathcal{P}_{tpc}$. •

To establish the correspondence of *FuTS* bisimilarity $\simeq_{tpc}$ for $\mathcal{S}_{tpc}$ of Definition 14 and timed bisimilarity $\sim_{tpc}$ for *TPC* of Definition 15, 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 $\rightsquigarrow$. The connection is established by Lemma 13. First we state an auxiliary result, which is commonly referred to as time-determinism (cf. [4]) and which can be shown straightforwardly by guarded induction.

Lemma 12. If $P \overset{n}{\rightsquigarrow} P'$ and $P \overset{n}{\rightsquigarrow} P''$, for $P, P', P'' \in \mathcal{P}_{tpc}$ and $n > 0$, then $P' = P''$. □

We use time-determinism of *TPC* in the proof of the following lemma.

Lemma 13.

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

(b) Let $P \in \mathcal{P}_{tpc}$. If $P \overset{\surd}{\rightarrowtail}_2 \mathcal{P}$ then $P \overset{n}{\rightsquigarrow} P' \iff \mathcal{P}(P') = \{n\}$.

Proof. Part (a) is similar to the corresponding part of Lemma 9. Part (b) can be shown by guarded induction for which we exhibit two cases (the others being similar or straightforward). For readability, we suppress the label $\surd$ of $\rightarrowtail_2$.

Case $(m).P$. Suppose $(m).P \rightarrowtail_2 \mathcal{P}$ and $P \rightarrowtail_2 \mathcal{P}'$. Then, by (TPF2), we have $\mathcal{P}(P') = \{\ell\}$, for $0 < \ell < m$, iff $P' = (m - \ell).P$, $\mathcal{P}(P') = \{m\}$ iff $P' = P$, and $\mathcal{P}(P') = \{\ell\}$, for $\ell > m$ iff $\mathcal{P}'(P') = \{\ell - m\}$. Now, if $(m).P \overset{n}{\rightsquigarrow} P'$ for $0 < n < m$, then $P' = (m - n).P$, because of rules (PRE) and (DEC) and Lemma 12. Therefore, $\mathcal{P}(P') = \mathcal{P}((m - n).P) = \{n\}$. If $(m).P \overset{n}{\rightsquigarrow} P'$ with $n = m$, then $P' = P$, as (PRE) applies (and with an appeal to Lemma 12). Therefore, $\mathcal{P}(P') = \mathcal{P}(P) = \{m\} = \{n\}$.

Finally, if $(m).P \overset{n}{\rightsquigarrow} P'$ for $n > m$, then we have $P \overset{n-m}{\rightsquigarrow} P'$, in view of rule (SUM) and because of time-determinism. By induction hypothesis, we obtain $\mathcal{P}'(P') = \{n - m\}$ and therefore $\mathcal{P}(P') = (m + \mathcal{P}')(P') = \{\, m + n \mid n \in \mathcal{P}'(P') \,\} = \{m + n - m\} = \{n\}$. Reversely, by rules (PRE) and (DEC) we have $(m).P \overset{\ell}{\rightsquigarrow} (m - \ell).P$, for $0 < \ell < m$ and $(m).P \overset{m}{\rightsquigarrow} P$. Moreover, if $\mathcal{P}(P') = \{\ell\}$, for $\ell > m$, then $\mathcal{P}'(P') = \{\ell - m\}$. By induction hypothesis, $P \overset{\ell-m}{\rightsquigarrow} P'$. Hence, $(m).P \overset{m+\ell-m}{\rightsquigarrow} P'$, i.e. $(m).P \overset{\ell}{\rightsquigarrow} P'$, by (SUM).

Case $P_1 + P_2$. Suppose $P_1 + P_2 \rightarrowtail_2 \mathcal{P}$. Then $\mathcal{P} = \mathcal{P}_1 \ [+] \ \mathcal{P}_2$ for $\mathcal{P}_1, \mathcal{P}_2 \in \mathcal{FS}(\mathcal{P}_{tpc}, \mathbf{2}^{\mathbb{N}})$ such that $P_1 \rightarrowtail_2 \mathcal{P}_1$ and $P_2 \rightarrowtail_2 \mathcal{P}_2$. If $P_1 + P_2 \overset{n}{\rightsquigarrow} P'$, then exist $P'_1, P'_2 \in \mathcal{P}_{tpc}$ such that $P_1 \overset{n}{\rightsquigarrow} P'_1$, $P_2 \overset{n}{\rightsquigarrow} P'_2$ and $P' = P'_1 + P'_2$, because (ALT) is the only rule applicable. By induction hypothesis, $\mathcal{P}_1(P'_1) = \{n\}$ and $\mathcal{P}_2(P'_2) = \{n\}$. Hence $\mathcal{P}(P') = (\mathcal{P}_1 \ [+] \ \mathcal{P}_2)(P'_1 + P'_2) = \{n\}$. In the other direction, if $\mathcal{P}(P') = \{n\}$, then $P' = P'_1 + P'_2$ for processes $P'_1, P'_2 \in \mathcal{P}_{tpc}$ such that $\mathcal{P}_1(P'_1) = \{n\}$ and $\mathcal{P}_2(P'_2) = \{n\}$. By induction hypothesis, $P_1 \overset{n}{\rightsquigarrow} P'_1$ and $P_2 \overset{n}{\rightsquigarrow} P'_2$, from which it follows that $P_1 + P_2 \overset{n}{\rightsquigarrow} P'_1 + P'_2$, i.e. $P_1 + P_2 \overset{n}{\rightsquigarrow} P'$, by (SUM). □

With Lemma 13 in place we are ready to show the correspondence of *FuTS* bisimilarity and timed bisimilarity for *TPC*.

Theorem 14. For any two processes $P_1, P_2 \in \mathcal{P}_{tpc}$ it holds that $P_1 \simeq_{tpc} P_2$ iff $P_1 \sim_{tpc} P_2$.

Proof. Suppose $P_1 \simeq_{tpc} P_2$, for $P_1, P_2 \in \mathcal{P}_{tpc}$. Let $R \subseteq \mathcal{P}_{tpc} \times \mathcal{P}_{tpc}$ be a bisimulation with respect to $\mathcal{S}_{tpc}$ such that $R(P_1, P_2)$. We verify that R meets the two transfer conditions of Definition 15.

If $P_1 \xrightarrow{a} Q_1$, for some $a \in \mathcal{A}$ and $Q_1 \in \mathcal{P}_{tpc}$, then $\theta_1(P_1)(a)(Q_1) = \mathit{true}$ by Lemma 13. From the definition of a *FuTS* bisimulation we obtain

$$\sum_{Q' \in [Q]_R} \theta_1(P_1)(a)(Q') = \sum_{Q' \in [Q]_R} \theta_1(P_2)(a)(Q') \tag{8.1}$$

for all $Q \in \mathcal{P}_{tpc}$. As we have seen before, we argue that summation of $\mathbb{B}$ is disjunction, and since $\theta_1(P_1)(a)(Q_1) = \mathit{true}$, there must exist $Q_2 \in [Q_1]_R$ such that $\theta_1(P_2)(Q_2) = \mathit{true}$. Hence, $R(Q_1, Q_2)$ and, by Lemma 13, $P_2 \xrightarrow{a} Q_2$.

If $P_1 \overset{n}{\rightsquigarrow} Q_1$, for some $n > 0$, then, by Lemma 13, $\theta_2(P_1)(\surd)(Q_1) = \{n\}$. From the definition of *FuTS* bisimulation we obtain

$$\sum_{Q' \in [Q]_R} \theta_2(P_1)(\surd)(Q') = \sum_{Q' \in [Q]_R} \theta_2(P_2)(\surd)(Q') \tag{8.2}$$

for all $Q \in \mathcal{P}_{tpc}$. Note, summation of the semiring $\mathbf{2}^{\mathbb{N}}$ is union of sets. So, by picking $Q = Q_1$ we have $n \in \sum_{Q' \in [Q_1]_R} \theta_2(P_2)(\surd)(Q')$. Thus, for some $Q_2 \in \mathcal{P}_{tpc}$ with $R(Q_1, Q_2)$ it holds that $n \in \theta_2(P_2)(\surd)(Q_2)$. It follows from Lemma 11b that $\theta_2(P_2)(\surd)(Q_2) = \{n\}$, and thus, again by Lemma 13, $P_2 \overset{n}{\rightsquigarrow} Q_2$.

Now suppose $P_1 \sim_{tpc} P_2$, for $P_1, P_2 \in \mathcal{P}_{tpc}$. Let $R \subseteq \mathcal{P}_{tpc} \times \mathcal{P}_{tpc}$ be a timed bisimulation such that $R(P_1, P_2)$. We verify that, with respect to P_1 and P_2, R meets the two summation conditions of Definition 8 for the case of $\mathcal{S}_{tpc}$, i.e., equations (8.1) and (8.2), for all $Q \in \mathcal{P}_{tpc}$ and $a \in \mathcal{A}$. We have

$$\begin{aligned}
&\textstyle\sum_{Q' \in [Q]_R} \theta_1(P_1)(a)(Q') \\
&\quad\Leftrightarrow\quad \exists Q' \in \mathcal{P}_{tpc} \colon R(Q', Q) \wedge \theta_1(P_1)(a)(Q') = \mathit{true} && \text{by structure of } \mathbb{B} \\
&\quad\Leftrightarrow\quad \exists Q' \in \mathcal{P}_{tpc} \colon R(Q', Q) \wedge P_1 \xrightarrow{a} Q' && \text{by Lemma 13} \\
&\quad\Leftrightarrow\quad \exists Q'' \in \mathcal{P}_{tpc} \colon R(Q'', Q) \wedge P_2 \xrightarrow{a} Q'' && R(P_1, P_2) \text{ and } R \text{ timed bisimulation} \\
&\quad\Leftrightarrow\quad \exists Q'' \in \mathcal{P}_{tpc} \colon R(Q'', Q) \wedge \theta_1(P_2)(a)(Q'') = \mathit{true} && \text{by Lemma 13} \\
&\quad\Leftrightarrow\quad \textstyle\sum_{Q'' \in [Q]_R} \theta_1(P_2)(a)(Q'') && \text{by structure of } \mathbb{B}
\end{aligned}$$

and also

$$
\begin{array}{lll}
n \in \sum_{Q' \in [Q]_R} \theta_2(P_1)(\surd)(Q') & & \\
\quad \Leftrightarrow \quad \exists Q' \in \mathcal{P}_{tpc} \colon R(Q', Q) \wedge n \in \theta_2(P_1)(\surd)(Q') & & \text{by structure of } \mathbf{2}^{\mathbb{N}} \\
\quad \Leftrightarrow \quad \exists Q' \in \mathcal{P}_{tpc} \colon R(Q', Q) \wedge \theta_2(P_1)(\surd)(Q') = \{n\} & & \text{by Lemma 11} \\
\quad \Leftrightarrow \quad \exists Q' \in \mathcal{P}_{tpc} \colon R(Q', Q) \wedge P_1 \overset{n}{\rightsquigarrow} Q' & & \text{by Lemma 13} \\
\quad \Leftrightarrow \quad \exists Q'' \in \mathcal{P}_{tpc} \colon R(Q'', Q) \wedge P_2 \overset{n}{\rightsquigarrow} Q'' & & R(P_1, P_2) \text{ and } R \text{ timed bisimulation} \\
\quad \Leftrightarrow \quad \exists Q'' \in \mathcal{P}_{tpc} \colon R(Q'', Q) \wedge \theta_2(P_2)(\surd)(Q'') = \{n\} & & \text{by Lemma 13} \\
\quad \Leftrightarrow \quad \exists Q'' \in \mathcal{P}_{tpc} \colon R(Q'', Q) \wedge n \in \theta_2(P_2)(\surd)(Q'') & & \text{by Lemma 11} \\
\quad \Leftrightarrow \quad \sum_{Q'' \in [Q]_R} \theta_2(P_2)(\surd)(Q'') & & \text{by structure of } \mathbf{2}^{\mathbb{N}}
\end{array}
$$

Thus, R satisfies the conditions for a *FuTS* bisimulation for $\mathcal{S}_{tpc}$. □

We conclude that also in the setting of a *FuTS* for discrete time involving the semiring $\mathbf{2}^{\mathbb{N}}$, we have an example of a correspondence result of *FuTS*-bisimilarity and bisimilarity based on a standard SOS definition. It is worth pointing out that in the proof above, the equivalence of $n \in \sum_{Q' \in [Q]_R} \theta_2(P_1)(\surd)(Q')$ and $\exists Q' \in \mathcal{P}_{tpc} \colon R(Q', Q) \wedge n \in \theta_2(P_1)(\surd)(Q')$, holds because we are working with a semiring of (finite) *sets* over $\mathbb{N}$ with summation to be interpreted as (finite) union. Was summation to be interpreted as sum over $\mathbb{N}$, as it would have been the case if we would have used the semiring $\mathbb{N}$, i.e. using $\mathcal{FS}(\mathcal{P}_{tpc}, \mathbb{N})$ instead of $\mathcal{FS}(\mathcal{P}_{tpc}, \mathbf{2}^{\mathbb{N}})$, then, from $n = \sum_{Q' \in [Q]_R} \theta_2(P_1)(\surd)(Q')$ we would not have been able to conclude $\exists Q' \in \mathcal{P}_{tpc} \colon R(Q', Q) \wedge n = \theta_2(P_1)(\surd)(Q')$, and vice-versa.

9. Nested *FuTS*

In this section we extend the applicability of the *FuTS* framework to more complex models, in particular those in which different aspects of behaviour are integrated in a non-orthogonal way—as it is the case for non-deterministic choice of probabilistic distributions over behaviour in probabilistic and Markov automata. We introduce the notion of a *nested FuTS*, namely a *FuTS* where the transition relation involves continuation functions that do not act on the set of states S directly, but instead on *functions* acting on S or, in the general case, on functions over the latter and so on. As mentioned in the introduction, here we restrict our investigation on nested *FuTS*s with two levels, namely nested *FuTS*s where the domain of the continuation functions is a set of functions the domain of which is the set S of states. In the following, we give the formal definition of a *simple* two-level nested *FuTS*, i.e. a nested *FuTS* involving two levels of continuations that has a single transition relation.

Definition 16. Let $\mathcal{L}$ be a set of labels and $\mathcal{R}_1$ and $\mathcal{R}_2$ be two semirings. A (simple) two-level nested *FuTS* $\mathcal{S}$, over $\mathcal{L}$ and $\mathcal{R}_1$ and $\mathcal{R}_2$ is a tuple $\mathcal{S} = (S, \rightarrowtail)$ with set of states S and transition relation $\rightarrowtail \subseteq S \times \mathcal{L} \times \mathcal{FS}(\mathcal{FS}(S, \mathcal{R}_1), \mathcal{R}_2)$. •

A two-level nested *FuTS* is called total and deterministic if, for all $s \in S$ and $\ell \in \mathcal{L}$, there exists exactly one $\psi \in \mathcal{FS}(\mathcal{FS}(S, \mathcal{R}_1), \mathcal{R}_2)$ such that $s \overset{\ell}{\rightarrowtail} \psi$. As before, for a total and deterministic nested *FuTS* we use the notation (S, θ) where the function θ has type $S \to \mathcal{L} \to \mathcal{FS}(\mathcal{FS}(S, \mathcal{R}_1), \mathcal{R}_2)$. Here, for $s \in S$, $\ell \in \mathcal{L}$, $\varphi \in \mathcal{FS}(S, \mathcal{R}_1)$, $y \in \mathcal{R}_2$, we have $\theta(s)(\ell)(\varphi) = y$ iff $\psi(\varphi) = y$ for the unique $\psi \in \mathcal{FS}(\mathcal{FS}(S, \mathcal{R}_1), \mathcal{R}_2)$ such that $s \overset{\ell}{\rightarrowtail} \psi$.

For a set of states S and a semiring $\mathcal{R}$, an equivalence relation R on S induces an equivalence relation on $\mathcal{FS}(S, \mathcal{R})$, referred to as the lifting of R to $\mathcal{FS}(S, \mathcal{R})$ and also denoted as R. The induced

relation R is defined by

$$R(\varphi_1, \varphi_2) \quad \text{iff} \quad \sum_{t' \in [t]_R} \varphi_1(t') = \sum_{t' \in [t]_R} \varphi_2(t') \quad \text{for all } t \in S$$

for $\varphi_1, \varphi_2 \in \mathcal{FS}(S, \mathcal{R})$. It is easy to see that R on $\mathcal{FS}(S, \mathcal{R})$ is indeed an equivalence relation. Therefore, the notion of a two-level bisimulation for a two-level nested *FuTS* given below is well-defined.

Definition 17. Let $\mathcal{S} = (S, \rightarrowtail)$ be a two-level nested *FuTS* over the label set $\mathcal{L}$ and semirings $\mathcal{R}_1$ and $\mathcal{R}_2$. An equivalence relation $R \subseteq S \times S$ is a two-level bisimulation for $\mathcal{S}$ if and only if $R(s_1, s_2)$ implies

$$\sum_{\varphi' \in [\varphi]_R} \theta(s_1)(\ell)(\varphi') = \sum_{\varphi' \in [\varphi]_R} \theta(s_2)(\ell)(\varphi') \tag{9.1}$$

for all $\ell \in \mathcal{L}$ and $\varphi \in \mathcal{FS}(S, \mathcal{R}_1)$. Two elements $s_1, s_2 \in S$ are called bisimilar for $\mathcal{S}$ if $R(s_1, s_2)$ for some two-level bisimulation R for $\mathcal{S}$. Notation $s_1 \simeq_{\mathcal{S}} s_2$. •

In Section 10 we will show that, in the setting of Markov Automata, the notion of a two-level bisimulation for a suitable two-level nested *FuTS* (having $\mathcal{R}_1 = \mathbb{R}_{\geqslant 0}$ and $\mathcal{R}_2 = \mathbb{B}$) coincides with the notion of strong bisimulation for Markov Automata.

As is to be expected, a total and deterministic two-level *FuTS* can be considered as a coalgebra of a suitable functor on sets.

Definition 18. Let $\mathcal{L}$ be a label set and $R = \langle \mathcal{R}_1, \mathcal{R}_2 \rangle$ be an pair of semirings. The functor $\mathcal{W}_R^{\mathcal{L}} : \mathbf{Set} \to \mathbf{Set}$ assigns to a set X the function space $\mathcal{FS}(\mathcal{FS}(X, \mathcal{R}_1), \mathcal{R}_2)^{\mathcal{L}}$ of all functions $\psi : \mathcal{L} \to \mathcal{FS}(\mathcal{FS}(X, \mathcal{R}_1), \mathcal{R}_2)$ and assigns to a mapping $f : X \to Y$ the mapping $\mathcal{W}_R^{\mathcal{L}}(f) : \mathcal{FS}(\mathcal{FS}(X, \mathcal{R}_1), \mathcal{R}_2)^{\mathcal{L}} \to \mathcal{FS}(\mathcal{FS}(Y, \mathcal{R}_1), \mathcal{R}_2)^{\mathcal{L}}$ where

$$\mathcal{W}_R^{\mathcal{L}}(f)(\Phi)(\ell)(\psi) = \sum_{\varphi \in \mathcal{FS}(f, \mathcal{R}_1)^{-1}(\psi)} \Phi(\ell)(\varphi)$$

for all $\Phi : \mathcal{L} \to \mathcal{FS}(\mathcal{FS}(X, \mathcal{R}_1), \mathcal{R}_2)$, $\ell \in \mathcal{L}$, $\psi \in \mathcal{FS}(Y, \mathcal{R}_1)$, where we use the function $\mathcal{FS}(f, \mathcal{R}_1) : \mathcal{FS}(X, \mathcal{R}_1) \to \mathcal{FS}(Y, \mathcal{R}_1)$ with $\mathcal{FS}(f, \mathcal{R}_1)(\varphi)(y) = \sum_{x \in f^{-1}(y)} \varphi(x)$ for $\varphi \in \mathcal{FS}(X, \mathcal{R}_1)$ and $y \in Y$. •

Note that in the definition above the sums exist since Φ and φ have finite support.

For readability we use $\mathcal{W}$ as shorthand for $\mathcal{W}_R^{\mathcal{L}}$, when the label set $\mathcal{L}$ and the pair of semirings R are clear from the context. It is readily checked that each $\mathcal{W}$ is a functor, in fact an accessible one being a composition of accessible functors. Thus, $\mathcal{W}$ possesses a final coalgebra. The associated notion of behavioural equivalence is denoted by $\approx_{\mathcal{W}}$. As before, we have for nested *FuTS* a correspondence result as well.

Theorem 15. Let $\mathcal{S} = (S, \theta)$ be a two-level nested *FuTS* over the label set $\mathcal{L}$ and the semirings $\mathcal{R}_1$ and $\mathcal{R}_2$. Let the functor $\mathcal{W}$ be as in Definition 18. Then $s_1 \simeq_{\mathcal{S}} s_2 \Leftrightarrow s_1 \approx_{\mathcal{W}} s_2$, for all $s_1, s_2 \in S$.

Proof. Let $s_1, s_2 \in S$. We first prove $s_1 \simeq_{\mathcal{S}} s_2 \Rightarrow s_1 \approx_{\mathcal{W}} s_2$. So, assume $s_1 \simeq_{\mathcal{S}} s_2$. Let $R \subseteq S \times S$ be a two-level bisimulation with $R(s_1, s_2)$. We turn the collection of equivalence classes S/R into a $\mathcal{W}$-coalgebra $\mathcal{S}_R = (S/R, \theta_R)$ by putting

$$\theta_R([s]_R)(\ell)(\bar{\varphi}) = \sum_{\varphi \in \mathcal{FS}(\varepsilon_R, \mathcal{R}_1)^{-1}(\bar{\varphi})} \theta(s)(\ell)(\varphi)$$

for $s \in S$, $\ell \in \mathcal{L}$, and $\bar{\varphi} \in \mathcal{FS}(S/R, \mathcal{R}_1)$ and $\varepsilon : S \to S/R$ the canonical mapping. This is well-defined since R is a two-level bisimulation and $\mathcal{FS}(\varepsilon_R, \mathcal{R}_1)^{-1}(\bar{\varphi})$ is an equivalence class of R, for all $\bar{\varphi} \in \mathcal{FS}(S/R, \mathcal{R}_1)$. For, if $\varphi_1, \varphi_2 \in \mathcal{FS}(\varepsilon_R, \mathcal{R}_1)^{-1}(\bar{\varphi})$, $t \in S$ then $\mathcal{FS}(\varepsilon_R, \mathcal{R}_1)(\varphi_1)([t]_R) = \mathcal{FS}(\varepsilon_R, \mathcal{R}_1)(\varphi_2)([t]_R)$. Thus $\sum_{t' \in [t]_R} \varphi_1(t') = \sum_{t' \in [t]_R} \varphi_2(t')$ for all $t \in S$, hence $R(\varphi_1, \varphi_2)$. Therefore, $\varepsilon_R : S \to S/R$ is a $\mathcal{W}$-homomorphism: for $\ell \in \mathcal{L}$ and $\bar{\varphi} \in \mathcal{FS}(S/R, \mathcal{R}_1)$, we have

$$
\begin{array}{lll}
\mathcal{W}(\varepsilon_R)(\,\theta(s)\,)(\ell)(\bar{\varphi}) & & \\
\quad = & \sum_{\varphi \in \mathcal{FS}(\varepsilon_R,\mathcal{R}_1)^{-1}(\bar{\varphi})} \theta(s)(\ell)(\varphi) & \text{by definition of } \mathcal{W} \\
\quad = & \theta_R(\,[s]_R\,)(\ell)(\,\bar{\varphi}\,) & \text{by definition of } \theta_R \\
\quad = & \theta_R(\,\varepsilon_R(s)\,)(\ell)(\,\bar{\varphi}\,) & \text{by definition of } \varepsilon_R
\end{array}
$$

Thus, $\mathcal{W}(\varepsilon_R) \circ \theta = \theta_R \circ \varepsilon_R$ and $\varepsilon_R : \mathcal{S} \to \mathcal{S}_R$ is a $\mathcal{W}$-homomorphism as claimed. Now, by uniqueness of a final morphism, we have $[\![\cdot]\!]^{\mathcal{S}}_{\mathcal{W}} = [\![\cdot]\!]^{\mathcal{S}_R}_{\mathcal{W}} \circ \varepsilon_R$. In particular, with respect to $\mathcal{S}$, this implies that $[\![s_1]\!]_{\mathcal{W}} = [\![s_2]\!]_{\mathcal{W}}$ since $\varepsilon_R(s_1) = \varepsilon_R(s_2)$. Thus, $s_1 \approx_{\mathcal{W}} s_2$ as was to be shown.

For the reverse, $s_1 \approx_{\mathcal{W}} s_2 \;\Rightarrow\; s_1 \simeq_{\mathcal{S}} s_2$, assume $s_1 \approx_{\mathcal{W}} s_2$, i.e. $[\![s_1]\!]_{\mathcal{W}} = [\![s_2]\!]_{\mathcal{W}}$, for $s_1, s_2 \in S$. Since the map $[\![\cdot]\!]_{\mathcal{W}} : (S, \theta) \to (\Omega, \omega)$ is a $\mathcal{W}$-homomorphism, the equivalence relation $R_{\mathcal{S}}$ given by $R_{\mathcal{S}}(s', s'') \Leftrightarrow [\![s']\!]_{\mathcal{W}} = [\![s'']\!]_{\mathcal{W}}$ is a two-level bisimulation: Suppose $R_{\mathcal{S}}(s', s'')$, i.e. $s' \approx_{\mathcal{W}} s''$, for some $s', s'' \in S$. Pick $\ell \in \mathcal{L}$, $t \in S$ and assume $[\![t]\!]_{\mathcal{W}} = w \in \Omega$. For $\mathcal{W}$ we have $\omega \circ [\![\cdot]\!]_{\mathcal{W}} = \mathcal{W}([\![\cdot]\!]_{\mathcal{W}}) \circ \theta$. Hence, for $s \in S$, $\ell \in \mathcal{L}$, $\chi \in \mathcal{FS}(\Omega, \mathcal{R}_1)$, it holds that

$$\omega(\,[\![s]\!]_{\mathcal{W}}\,)(\ell)(\chi) = \mathcal{W}(\,[\![\cdot]\!]_{\mathcal{W}}\,)(\theta(s))(\ell)(\chi) = \sum_{\varphi \in \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}},\mathcal{R}_1)^{-1}(\chi)} \theta(s)(\ell)(\varphi) \tag{9.2}$$

Moreover, we have, for $\varphi_1, \varphi_2 \in \mathcal{FS}(S, \mathcal{R}_1)$, that

$$R_S(\,\varphi_1,\, \varphi_2\,) \iff \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}}, \mathcal{R}_1)(\varphi_1) = \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}}, \mathcal{R}_1)(\varphi_2)$$

since we observe that

$$
\begin{array}{ll}
\mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}}, \mathcal{R}_1)(\varphi_1) = \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}}, \mathcal{R}_1)(\varphi_2) & \\
\quad \Leftrightarrow & \forall w \in [\![S]\!]_{\mathcal{W}} \colon \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}}, \mathcal{R}_1)(\varphi_1)(w) = \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}}, \mathcal{R}_1)(\varphi_2)(w) \\
& \text{since both } \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}}, \mathcal{R}_1)(\varphi_1)(w), \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}}, \mathcal{R}_1)(\varphi_2)(w) = 0 \text{ if } [\![\cdot]\!]_{\mathcal{W}}{}^{-1}(w) = \varnothing \\
\quad \Leftrightarrow & \forall t \in S : \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}}, \mathcal{R}_1)(\varphi_1)([\![t]\!]_{\mathcal{W}}) = \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}}, \mathcal{R}_1)(\varphi_2)([\![t]\!]_{\mathcal{W}}) \\
\quad \Leftrightarrow & \forall t \in S : \sum_{t' \in [\![\cdot]\!]_{\mathcal{W}}{}^{-1}([\![t]\!]_{\mathcal{W}})} \varphi_1(t') = \sum_{t' \in [\![\cdot]\!]_{\mathcal{W}}{}^{-1}([\![t]\!]_{\mathcal{W}})} \varphi_2(t') \\
& \text{by definition of } \mathcal{FS}(\cdot, \mathcal{R}_1) \\
\quad \Leftrightarrow & \forall t \in S : \sum_{t' \in [t]_{R_{\mathcal{S}}}} \varphi_1(t') = \sum_{t' \in [t]_{R_{\mathcal{S}}}} \varphi_2(t') \\
& \text{since } t' \in [t]_{R_{\mathcal{S}}} \text{ iff } [\![t']\!]_{\mathcal{W}} = [\![t]\!]_{\mathcal{W}} \\
\quad \Leftrightarrow & R_{\mathcal{S}}(\varphi_1, \varphi_2) \\
& \text{by definition of } R_{\mathcal{S}} \text{ on } \mathcal{FS}(S, \mathcal{R}_1)
\end{array}
$$

Therefore,

$$\varphi' \in [\varphi]_{R_{\mathcal{S}}} \iff \varphi' \in \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}}, \mathcal{R}_1)^{-1}(\chi) \qquad \text{for } \chi = \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}}, \mathcal{R}_1)(\varphi) \tag{9.3}$$

Now, let $s', s'' \in S$ such that $R_{\mathcal{S}}(s', s'')$, and choose any $\ell \in \mathcal{L}$ and $\varphi \in \mathcal{FS}(S, \mathcal{R}_1)$. Put $\chi = \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}}, \mathcal{R}_1)(\varphi)$. Then we have

$$
\begin{array}{lll}
\sum_{\varphi' \in [\varphi]_{R_{\mathcal{S}}}} \theta(s')(\ell)(\varphi') & & \\
\quad = & \sum_{\varphi' \in \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}},\mathcal{R}_1)^{-1}(\chi)} \theta(s')(\ell)(\varphi') & \text{by Equation (9.3 and definition } \chi) \\
\quad = & \omega(\,[\![s']\!]_{\mathcal{W}}\,)(\ell)(\chi) & \text{by Equation (9.2)} \\
\quad = & \omega(\,[\![s'']\!]_{\mathcal{W}}\,)(\ell)(\chi) & s' \approx_{\mathcal{W}} s'' \text{ by assumption} \\
\quad = & \sum_{\varphi' \in \mathcal{FS}(\,[\![\cdot]\!]_{\mathcal{W}},\mathcal{R}_1)^{-1}(\chi)} \theta(s'')(\ell)(\varphi') & \text{by Equation (9.2)} \\
\quad = & \sum_{\varphi' \in [\varphi]_{R_{\mathcal{S}}}} \theta(s'')(\ell)(\varphi') & \text{by Equation (9.3 and definition } \chi)
\end{array}
$$

Thus, if $R_{\mathcal{S}}(s', s'')$ then $\sum_{\varphi' \in [\varphi]_{R_{\mathcal{S}}}} \theta(s')(\ell)(\varphi') = \sum_{\varphi' \in [\varphi]_{R_{\mathcal{S}}}} \theta(s'')(\ell)(\varphi')$ for all $\ell \in \mathcal{L}$ and $\varphi \in \mathcal{FS}(S, \mathcal{R}_1)$. Therefore, $R_{\mathcal{S}}$ is a two-level bisimulation according to Definition 17. Since $[\![s_1]\!]_{\mathcal{W}} =$

$[\![s_2]\!]_{\mathcal{W}}$, it follows that $R_{\mathcal{S}}(s_1, s_2)$. Thus $R_{\mathcal{S}}$ is a two-level bisimulation relating s_1 and s_2. Conclusion, it holds that $s_1 \simeq_{\mathcal{S}} s_2$. □

Above we introduced the notion of a two-level nested *FuTS* and an associated notion of bisimulation. Also in the case of such nested *FuTS*, *FuTS*-bisimulation and behavioral equivalence of the corresponding functor coincides. Combination of nested *FuTS*, or combination of nested and simple *FuTS*, over the same set of states, is a straightforward generalization along the lines of Section 6. We will not pursue unfolding of the details here. In the next section we will encounter an example of such a construction.

10. *FuTS* Semantics of a language for Markov Automata

As a final application of the *FuTS* approach to modeling quantitative behaviour we consider so-called Markov automata (MA). A Markov automaton, as proposed in [22, 23, 53], combines non-deterministic and probabilistic behaviour, on the one hand, with stochastic time behaviour, on the other hand. Therefore, we need a combination of a nested and a simple *FuTS* to model the respective behaviour.

The definition of an MA here follows [53]. We first recall some definitions from [53, 20] with $\mathit{Distr}(S) \subseteq \mathcal{FS}(S, \mathbb{R}_{\geqslant 0})$ denoting the class of (finitely supported) probability distributions over S.

The superposition of non-deterministic and probabilistic behaviour is provided in Markov automata by means of a combination of a standard choice operator ' + ' together with the probabilistic extension of action prefix $a.\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\}$ for $a \in \mathcal{A}$, $h > 0$, and $p_1, \ldots, p_h \in (0, 1]$ such that $p_1 + \cdots + p_h = 1$. The syntactic construct $\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\}$ denotes the distribution $\mu_{\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\}}$ over processes defined by

$$\mu_{\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\}} = \textstyle\sum_{i=1}^{h} [P_i \mapsto p_i]$$

The intuitive meaning is then obvious: process $a.\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\}$ performs action a and then behaves as process P with probability $\mu_{\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\}}(P)$.

A process language for Markov Automata called MAPA (Markov Automata Process Algebra) has been proposed in [53, 54, 52]. MAPA includes a rich data system and is equipped with restrictions to facilitate state space generation of relatively small models. Below, we consider *MAL* as introduced in [18]. *MAL* constitutes a simplified fragment of MAPA which highlights how nested non-deterministic and probabilistic behaviour combined with Markovian behaviour can be modeled in the *FuTS* framework.

Definition 19. The set $\mathcal{P}_{mal}$ of *MA* processes is given by the grammar

$$P \;::=\; \mathbf{nil} \mid a.\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\} \mid \lambda.P \mid P + P \mid P \parallel_A P \mid X$$

where a ranges over the set of actions $\mathcal{A}$, p_i over the interval $(0, 1]$, λ over $\mathbb{R}_{>0}$, A over the set of finite subsets of $\mathcal{A}$ and X over the set of constants $\mathcal{X}$. For an probabilistic action-prefix $a.\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\}$ it is required that $h > 0$ and $p_1 + \cdots + p_h = 1$. •

We assume the same notation, guardedness requirements and conventions for constant definitions as in Section 5 for *PEPA*, *IML* and *TPC*.

In the setting of $\mathcal{P}_{mal}$ we use $\mathscr{P}, \mathscr{Q}$ to range over $\mathcal{FS}(\,\mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0}), \mathbb{B})$ and $\mathcal{P}, \mathcal{Q}$ to range over $\mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0})$. We use μ, ν to range over $\mathit{Distr}(\mathcal{P}_{mal}) \subseteq \mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0})$. As before, we let $\mathcal{P}_1 + \mathcal{P}_2$

$$\text{(NIL1)}\ \frac{a \in \mathcal{A}}{\mathbf{nil} \overset{a}{\rightarrowtail}_1 [\,]_{\mathbb{B}}} \qquad \text{(NIL2)}\ \frac{}{\mathbf{nil} \overset{\delta}{\rightarrowtail}_2 [\,]_{\mathbb{R}_{\geqslant 0}}} \qquad \text{(RPF1)}\ \frac{a \in \mathcal{A}}{\lambda.P \overset{a}{\rightarrowtail}_1 [\,]_{\mathbb{B}}} \qquad \text{(RPF2)}\ \frac{}{\lambda.P \overset{\delta}{\rightarrowtail}_2 [P \mapsto \lambda]}$$

$$\text{(APF1)}\ \frac{}{a.\{\, p_1{\cdot}P_1 \,\Box\, \cdots \,\Box\, p_h{\cdot}P_h \,\} \overset{a}{\rightarrowtail}_1 [\mu_{\{\, p_1{\cdot}P_1 \,\Box\cdots\Box\, p_h{\cdot}P_h \,\}} \mapsto \mathit{true}]}$$

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

$$\text{(CHO1)}\ \frac{P \overset{a}{\rightarrowtail}_1 \mathscr{P} \quad Q \overset{a}{\rightarrowtail}_1 \mathscr{Q}}{P + Q \overset{a}{\rightarrowtail}_1 \mathscr{P} + \mathscr{Q}} \qquad \text{(CHO2)}\ \frac{P \overset{\delta}{\rightarrowtail}_2 \mathcal{P} \quad Q \overset{\delta}{\rightarrowtail}_2 \mathcal{Q}}{P + Q \overset{\delta}{\rightarrowtail}_2 \mathcal{P} + \mathcal{Q}}$$

$$\text{(PAR1)}\ \frac{P \overset{a}{\rightarrowtail}_1 \mathscr{P} \quad Q \overset{a}{\rightarrowtail}_1 \mathscr{Q} \quad a \notin A}{P \parallel_A Q \overset{a}{\rightarrowtail}_1 (\mathscr{P} \parallel_A \mathbf{D}_Q) + (\mathbf{D}_P \parallel_A \mathscr{Q})} \qquad \text{(PAR2)}\ \frac{P \overset{a}{\rightarrowtail}_1 \mathscr{P} \quad Q \overset{a}{\rightarrowtail}_1 \mathscr{Q} \quad a \in A}{P \parallel_A Q \overset{a}{\rightarrowtail}_1 \mathscr{P} \parallel_A \mathscr{Q}}$$

$$\text{(PAR3)}\ \frac{P \overset{\delta}{\rightarrowtail}_2 \mathcal{P} \quad Q \overset{\delta}{\rightarrowtail}_2 \mathcal{Q}}{P \parallel_A Q \overset{\delta}{\rightarrowtail}_2 (\mathcal{P} \parallel_A \mathbf{D}_Q) + (\mathbf{D}_P \parallel_A \mathcal{Q})}$$

$$\text{(CON1)}\ \frac{P \overset{a}{\rightarrowtail}_1 \mathscr{P} \quad X := P}{X \overset{a}{\rightarrowtail}_1 \mathscr{P}} \qquad \text{(CON2)}\ \frac{P \overset{\delta}{\rightarrowtail}_2 \mathcal{P} \quad X := P}{X \overset{\delta}{\rightarrowtail}_2 \mathcal{P}}$$

Figure 8: *FuTS* Transition Deduction System for *MAL*.

be the pointwise sum of $\mathcal{P}_1$ and $\mathcal{P}_2$. (Note, we are adding rates here.) We put $\mathbf{D}_P = [P \mapsto 1]$ in $\mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0})$ and define $\mathcal{P}_1 \parallel_A \mathcal{P}_2 : \mathcal{P}_{mal} \to \mathbb{R}_{\geqslant 0}$, for $\mathcal{P}_1, \mathcal{P}_2 \in \mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0})$ and $A \subseteq \mathcal{A}$, by

$$(\mathcal{P}_1 \parallel_A \mathcal{P}_2)(R) = \begin{cases} \mathcal{P}_1(R_1) \cdot \mathcal{P}_2(R_2) & \text{if } R = R_1 \parallel_A R_2 \text{ for some } R_1, R_2 \in \mathcal{P}_{mal} \\ 0 & \text{otherwise} \end{cases}$$

Note $\mathcal{P}_1 \parallel_A \mathcal{P}_2 \in \mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0})$. Moreover, if $\mu_1, \mu_2 \in \mathit{Distr}(\mathcal{P}_{mal})$ then $\mu_1 \parallel_A \mu_2 \in \mathit{Distr}(\mathcal{P}_{mal})$ too, since

$$\textstyle\sum_{R \in \mathcal{P}_{mal}} (\mu_1 \parallel_A \mu_2)(R) = \sum_{R_1, R_2 \in \mathcal{P}_{mal}} \mu_1(R_1) \cdot \mu_2(R_2) = \big(\sum_{R_1 \in \mathcal{P}_{mal}} \mu_1(R_1)\big) \cdot \big(\sum_{R_2 \in \mathcal{P}_{mal}} \mu_2(R_2)\big)$$

while $\sum_{R_1 \in \mathcal{P}_{mal}} \mu_1(R_1), \sum_{R_2 \in \mathcal{P}_{mal}} \mu_2(R_2) = 1$. For $\mathscr{P}_1, \mathscr{P}_2 \in \mathcal{FS}(\,\mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0}), \mathbb{B})$ and $A \subseteq \mathcal{A}$, we also use constructs $\mathscr{P}_1 + \mathscr{P}_2$ and $\mathscr{P}_1 \parallel_A \mathscr{P}_2$ where $(\mathscr{P}_1 + \mathscr{P}_2)(\mu) = \mathscr{P}_1(\mu) \vee \mathscr{P}_2(\mu)$ is pointwise disjunction, and $\mathscr{P}_1 \parallel_A \mathscr{P}_2$ is defined by

$$\textstyle\sum_{\mu_1, \mu_2 \colon \mathscr{P}_1(\mu_1) = \mathit{true} \,\wedge\, \mathscr{P}_2(\mu_2) = \mathit{true}} [\mu_1 \parallel_A \mu_2 \mapsto \mathit{true}]$$

Thus $(\mathscr{P}_1 \parallel_A \mathscr{P}_2)(\mu) = \mathit{true}$ iff $\mu = \mu_1 \parallel_A \mu_2$, for μ_1 such that $\mathscr{P}_1(\mu_1) = \mathit{true}$ and μ_2 such that $\mathscr{P}_2(\mu_2) = \mathit{true}$. We overload $\mathbf{D}_P$ for $P \in \mathcal{P}_{mal}$; with respect to $\mathcal{FS}(\,\mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0}), \mathbb{B})$ we have $\mathbf{D}_P = [[P \mapsto 1] \mapsto \mathit{true}]$. Because of the contexts no confusion arises whether to interpret $\mathbf{D}_P$ with respect to $\mathcal{FS}(\,\mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0}), \mathbb{B})$ or with respect to $\mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0})$.

With the operators defined above in place, and a combined treatment of actions and probabilities vs. stochastic delays, it is straightforward to capture the semantics of *MAL* with *FuTS*, cf. [18].

Definition 20. The formal semantics of $\mathcal{P}_{mal}$ is given by the *FuTS* $\mathcal{S}_{mal} = (\mathcal{P}_{mal}, \rightarrowtail_1, \rightarrowtail_2)$, a general *FuTS* over the label sets $\mathcal{A}$ and $\Delta = \{\delta\}$ and the semirings $\mathbb{R}_{\geqslant 0}$, $\mathbb{B}$ and $\mathbb{R}_{\geqslant 0}$ again with transition relations $\rightarrowtail_1 \subseteq \mathcal{P}_{mal} \times \mathcal{A} \times \mathcal{FS}(\,\mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0}), \mathbb{B})$ and $\rightarrowtail_2 \subseteq \mathcal{P}_{mal} \times \Delta \times \mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0})$ defined as the least relations satisfying the rules of Figure 8. •

$$\text{(ACT)}\ \frac{}{a.[\,p_1{\cdot}P_1 \oplus \cdots \oplus p_h{\cdot}P_h\,] \xrightarrow{a} \mu_{[\,p_1\cdot P_1\oplus\cdots\oplus p_h\cdot P_h\,]}} \qquad \text{(DELAY)}\ \frac{}{\lambda.P \xrightarrow{\lambda} P}$$

$$\text{(CHO1)}\ \frac{P \xrightarrow{a} \mu}{P+Q \xrightarrow{a} \mu} \qquad \text{(CHO2)}\ \frac{Q \xrightarrow{a} \nu}{P+Q \xrightarrow{a} \nu} \qquad \text{(CHO3)}\ \frac{P \xrightarrow{\lambda}_2 P'}{P+Q \xrightarrow{\lambda}_2 P'} \qquad \text{(CHO4)}\ \frac{Q \xrightarrow{\lambda}_2 Q'}{P+Q \xrightarrow{\lambda}_2 Q'}$$

$$\text{(PAR1)}\ \frac{P \xrightarrow{a} \mu \quad a \notin A}{P \parallel_A Q \xrightarrow{a} \mu \parallel_A \mathbf{D}_Q} \qquad \text{(PAR2)}\ \frac{Q \xrightarrow{a} \nu \quad a \notin A}{P \parallel_A Q \xrightarrow{a} \mathbf{D}_P \parallel_A \nu} \qquad \text{(PAR3)}\ \frac{P \xrightarrow{a} \mu \quad Q \xrightarrow{a} \nu \quad a \in A}{P \parallel_A Q \xrightarrow{a} \mu \parallel_A \nu}$$

$$\text{(PAR4)}\ \frac{P \xrightarrow{\lambda} P'}{P \parallel_A Q \xrightarrow{\lambda} P' \parallel_A Q} \qquad \text{(PAR5)}\ \frac{Q \xrightarrow{\lambda} Q'}{P \parallel_A Q \xrightarrow{\lambda} P \parallel_A Q'}$$

$$\text{(REC1)}\ \frac{P \xrightarrow{a} \mu \quad X := P}{X \xrightarrow{a} \mu} \qquad \text{(REC2)}\ \frac{P \xrightarrow{\lambda} P' \quad X := P}{X \xrightarrow{\lambda} P'}$$

Figure 9: Standard Transition Deduction System for *MAL*.

By guarded induction we obtain that the finitely supported functions involved in the definition of $\rightarrowtail_1$ are indeed probability distributions. Ultimately this relies on the restriction on the extended prefix, for the process $a.\{\, p_1{\cdot}P_1 \,\Box\, \cdots \,\Box\, p_h{\cdot}P_h \,\}$ the finite sum $p_1 + \cdots + p_h$ must be equal to 1.

Lemma 16. For all $P \in \mathcal{P}_{mal}, a \in \mathcal{A}, \mathscr{P} \in \mathcal{FS}(\,\mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0})\,, \mathbb{B})$ and $\mathcal{P} \in \mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0})$ if $P \overset{a}{\rightarrowtail}_1 \mathscr{P}$ and $\mathscr{P}(\mathcal{P}) = \mathit{true}$, then $\mathcal{P} \in \mathit{Distr}(\mathcal{P}_{mal})$. □

It is not difficult either to verify that $\mathcal{S}_{mal}$ is a total and deterministic combined *FuTS*, i.e. for $P \in \mathcal{P}_{mal}$, $a \in \mathcal{A}$ we have $P \overset{a}{\rightarrowtail}_1 \mathscr{P}$ for exactly one $\mathscr{P} \in \mathcal{FS}(\,\mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0})\,, \mathbb{B})$ and $P \overset{\delta}{\rightarrowtail}_2 \mathcal{P}$ for exactly one $\mathcal{P} \in \mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0})$.

Lemma 17. The general *FuTS* $\mathcal{S}_{mal}$ is total and deterministic. □

Below we use $\mathcal{S}_{mal} = (\mathcal{P}_{mal}, \theta_1, \theta_2)$ with $\theta_1 : \mathcal{P}_{mal} \to \mathcal{FS}(\mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}), \mathbb{B})$ and $\theta_2 : \mathcal{P}_{mal} \to \mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R})$ induced by $\rightarrowtail_1$ and $\rightarrowtail_2$, respectively. We write $\simeq_{mal}$ for the associated notion of bisimilarity. Recall, for θ_1 the relevant definition is Definition 17 on page 27, while for θ_2 we of course refer to Definition 2 of page 7, as shown below, for clarity.

Definition 21. An equivalence relation $R \subseteq \mathcal{P}_{mal} \times \mathcal{P}_{mal}$ is an $\mathcal{S}_{mal}$-bisimulation if and only if R is a nested bisimulation with respect to θ_1 and a simple bisimulation with respect to θ_2. •

If we unfold the definitions for the two types of *FuTS* bisimulation we obtain that an equivalence relation $R \subseteq \mathcal{P}_{mal} \times \mathcal{P}_{mal}$ is an $\mathcal{S}_{mal}$-bisimulation, if for all $P_1, P_2 \in \mathcal{P}_{mal}$ such that $R(P_1, P_2)$, it holds that

- for all $a \in \mathcal{A}$ and $\mu \in \mathit{Distr}(\mathcal{P}_{mal})$: $\sum_{\mu' \in [\mu]_R} \theta_1(P_1)(a)(\mu') = \sum_{\mu' \in [\mu]_R} \theta_1(P_2)(a)(\mu')$, and
- for all $Q \in \mathcal{P}_{mal}$: $\sum_{Q' \in [Q]_R} \theta_2(P_1)(\delta)(Q') = \sum_{Q' \in [Q]_R} \theta_2(P_2)(\delta)(Q')$

with R on $\mathit{Distr}(\mathcal{P}_{mal})$ induced by R on $\mathcal{P}_{mal}$. Recall that, for $\mu_1, \mu_2 \in \mathit{Distr}(\mathcal{P}_{mal})$, $R(\mu_1, \mu_2)$ if and only if $\sum_{Q' \in [Q]_R} \mu_1(Q') = \sum_{Q' \in [Q]_R} \mu_2(Q')$ for all $Q \in \mathcal{P}_{mal}$.

A standard LTS-based operational semantics of *MAL* is given by the SOS rules of Figure 9. The semantics is the similar to the one reported in [53, 54]. Here, however, the technical overhead of decorations on transitions as used in the above mentioned papers is avoided at the expense of implicit multiplicities, in line with the treatment of *PEPA* and *IML* in Sections 5 and 7, respectively. Note, as *MAL* extends *IML*, there are separate rules for interactive transitions (ACT, CHO1–2,

PAR1–3 and REC1) captured by the transition relation $\longrightarrow$, and for Markovian transitions (DELAY, CHO3–4, PAR4–5, REC2) captured by the transition relation $\dashrightarrow$.

Definition 22. The semantics of the process language *MAL* is the four-tuple $(\mathcal{P}_{mal}, \mathcal{A}, \longrightarrow, \dashrightarrow)$ where the probabilistic transition relation $\longrightarrow \subseteq \mathcal{P}_{mal} \times \mathcal{A} \times \mathit{Distr}(\mathcal{P}_{mal})$ and the standard transition relation $\dashrightarrow \subseteq \mathcal{P}_{mal} \times \mathbb{R}_{>0} \times \mathcal{P}_{mal}$ are given by the SOS rules of Figure 9. •

Similar to our treatment of $\mathcal{P}_{iml}$ in Section 7, we introduce the functions $\mathbf{I}$ and $\mathbf{M}$ based on the transition relations $\longrightarrow$ and $\dashrightarrow$ of Definition 22 for $\mathcal{P}_{mal}$. Now, for the interactive part of *MAL*, we have $\mathbf{I}\colon \mathcal{P}_{mal} \times \mathcal{A} \times \mathbf{2}^{\mathit{Distr}(\mathcal{P}_{mal})} \to \mathbb{B}$ given by $\mathbf{I}(P, a, \mathscr{C}) = \mathit{true}$ if the set $\{\mu \in \mathscr{C} \mid P \xrightarrow{a} \mu\}$ is non-empty, for all $P \in \mathcal{P}_{mal}$, $a \in \mathcal{A}$ and any subset $\mathscr{C} \subseteq \mathit{Distr}(\mathcal{P}_{mal})$. The Markovian part of *MAL* is similar to that of *IML*. We define for *MAL* the function $\mathbf{M}\colon \mathcal{P}_{mal} \times \mathcal{P}_{mal} \to \mathbb{R}_{\geqslant 0}$ by $\mathbf{M}(P, P') = \sum \{\!|\, \lambda \mid P \overset{\lambda}{\dashrightarrow} P' \,|\!\}$. Because of the implicit multiplicities of the SOS of Definition 22, the comprehension is over the multiset of transitions leading from P to P' with label λ. We also extend $\mathbf{M}$, now to $\mathcal{P}_{mal} \times \mathbf{2}^{\mathcal{P}_{mal}}$, by $\mathbf{M}(P, C) = \sum_{P' \in C} \sum \{\!|\, \lambda \mid P \overset{\lambda}{\dashrightarrow} P' \,|\!\}$, for $P \in \mathcal{P}_{mal}$ and $C \subseteq \mathcal{P}_{mal}$. With the adapted functions $\mathbf{I}$ and $\mathbf{M}$ in place, the notion of strong bisimulation for *MAL* is defined as follows.

Definition 23. An equivalence relation $R \subseteq \mathcal{P}_{mal} \times \mathcal{P}_{mal}$ is called a strong bisimulation for *MAL* if, for all $P_1, P_2 \in \mathcal{P}_{mal}$ such that $R(P_1, P_2)$, it holds that

- for all $a \in \mathcal{A}$ and $\mu \in \mathit{Distr}(\mathcal{P}_{mal})$: $\mathbf{I}(P_1, a, [\mu]_R) \iff \mathbf{I}(P_2, a, [\mu]_R)$
- for all $Q \in \mathcal{P}_{mal}$: $\mathbf{M}(P_1, [Q]_R) = \mathbf{M}(P_2, [Q]_R)$

with the relation R on $\mathit{Distr}(\mathcal{P}_{mal})$ induced by the relation R on $\mathcal{P}_{mal}$. Two processes $P_1, P_2 \in \mathcal{P}_{mal}$ are called strongly bisimilar if it holds that $R(P_1, P_2)$ for a strong bisimulation R for *MAL*, notation $P_1 \sim_{ma} P_2$. •

Recall, again, that the relation $R \subseteq \mathcal{P}_{mal} \times \mathcal{P}_{mal}$ induces relation $R \subseteq \mathit{Distr}(\mathcal{P}_{mal}) \times \mathit{Distr}(\mathcal{P}_{mal})$ by $R(\mu_1, \mu_2)$ if and only if $\sum_{Q' \in [Q]_R} \mu_1(Q') = \sum_{Q' \in [Q]_R} \mu_2(Q')$ for all $Q \in \mathcal{P}_{mal}$. In line with what we have seen in the previous sections, the crux for relating the notion of $\mathcal{S}_{mal}$-bisimulation and the notion of strong bisimulation of Definition 23 is the following result.

Lemma 18.

(a) Let $P \in \mathcal{P}_{mal}$ and $a \in \mathcal{A}$. If $P \overset{a}{\rightarrowtail}_1 \mathscr{P}$ then $P \xrightarrow{a} \mu \iff \mathscr{P}(\mu) = \mathit{true}$.

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

Proof. (a) Guarded induction. Let $a \in \mathcal{A}$. We treat the cases $a.\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\}$ and $P_1 \parallel_A P_2$ for $a \in A$.

Case $a.\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\}$. $a.\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\} \overset{a}{\rightarrowtail}_1 [\mu_{\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\}} \mapsto \mathit{true}]$, while $a.\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\} \xrightarrow{a} \mu_{\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\}}$ is the only transition for $a.\{\, p_1 \cdot P_1 \,\Box \cdots \Box\, p_h \cdot P_h \,\}$.

Case $P_1 \parallel_A P_2$, $a \in A$. Assume $P_1 \parallel_A P_2 \overset{a}{\rightarrowtail}_1 \mathscr{P}$. Then $\mathscr{P} = \mathscr{P}_1 \parallel_A \mathscr{P}_2$ for $\mathscr{P}_1, \mathscr{P}_2 : \mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0}) \to \mathbb{B}$ such that $P_1 \overset{a}{\rightarrowtail}_1 \mathscr{P}_1$, $P_2 \overset{a}{\rightarrowtail}_1 \mathscr{P}_2$. Suppose $P_1 \parallel_A P_2 \xrightarrow{a} \mu$. Then there exist $\mu_1, \mu_2 \in \mathit{Distr}(\mathcal{P}_{mal})$ such that $P_1 \xrightarrow{a} \mu_1$, $P_2 \xrightarrow{a} \mu_2$ and $\mu = \mu_1 \parallel_A \mu_2$, since only rule (PAR3) of Figure 9 applies. By induction hypothesis, $\mathscr{P}_1(\mu_1) = \mathit{true}$ and $\mathscr{P}_2(\mu_2) = \mathit{true}$. Hence $\mathscr{P}(\mu) = (\mathscr{P}_1 \parallel_A \mathscr{P}_2)(\mu_1 \parallel_A \mu_2) = \mathit{true}$ by definition of $\parallel_A$ on $\mathcal{FS}(\mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0}), \mathbb{B})$. Reversely, suppose $\mathscr{P}(\mu) = \mathit{true}$. Then $\mu = \mu_1 \parallel_A \mu_2$ for $\mu_1, \mu_2 \in \mathit{Distr}(\mathcal{P}_{mal})$ such that $\mathscr{P}_1(\mu_1) = \mathit{true}$ and $\mathscr{P}_2(\mu_2) = \mathit{true}$. By induction hypothesis, $P_1 \xrightarrow{a} \mu_1$ and $P_2 \xrightarrow{a} \mu_2$. Hence $P_1 \parallel_A P_2 \xrightarrow{a} \mu_1 \parallel_A \mu_2$ by rule (PAR3), i.e. $P_1 \parallel_A P_2 \xrightarrow{a} \mu$.

The other cases are left to the reader.

(b) Guarded induction. Compared to the proof of Lemma 9 there is only one new case, viz. for processes of the form $a.\{\, p_1 \cdot P_1 \,\Box\, \cdots \,\Box\, p_h \cdot P_h \,\}$. This case is straightforward, since, on the one hand, $a.\{\, p_1 \cdot P_1 \,\Box\, \cdots \,\Box\, p_h \cdot P_h \,\} \overset{\delta}{\rightarrowtail}_2 [\,]_{\mathbb{R}_{\geqslant 0}}$ by definition of $\overset{\delta}{\rightarrowtail}_2$ and, on the other hand, we have that $a.\{\, p_1 \cdot P_1 \,\Box\, \cdots \,\Box\, p_h \cdot P_h \,\} \overset{\lambda}{\dashrightarrow} P'$ for *no* $P' \in \mathcal{P}_{mal}$ by definition of $\dashrightarrow$.

The remaining cases are similar to the proof for the corresponding lemma for *IML* and left to the reader. □

We are now in a position to relate the notions of *FuTS* bisimilarity $\simeq_{mal}$ and standard strong bisimilarity $\sim_{ma}$ for *MAL*.

Theorem 19. For any two processes $P_1, P_2 \in \mathcal{P}_{mal}$ it holds that $P_1 \simeq_{mal} P_2$ iff $P_1 \sim_{ma} P_2$.

Proof. Let R be an equivalence relation on $\mathcal{P}_{mal}$. Pick $P \in \mathcal{P}_{mal}$, $a \in \mathcal{A}$ and choose any $\mathscr{P} \in \mathcal{FS}(\mathcal{FS}(\mathcal{P}_{mal}, \mathbb{R}_{\geqslant 0}), \mathbb{B})$. Suppose $P \overset{a}{\rightarrowtail}_1 \mathscr{P}$. Thus $\theta_1(P)(a) = \mathscr{P}$. Then we have, for any $\mu \in \mathit{Distr}(\mathcal{P}_{mal})$,

$$\begin{array}{lcll} \mathbf{I}(P, a, [\mu]_R) & \Leftrightarrow & \exists \mu' \in [\mu]_R \colon P \xrightarrow{a} \mu' & \text{by definition of } \mathbf{I} \\ & \Leftrightarrow & \exists \mu' \in [\mu]_R \colon \mathscr{P}(\mu') = \mathit{true} & \text{by Lemma 18a} \\ & \Leftrightarrow & \sum_{\mu' \in [\mu]_R} \theta_1(P)(a)(\mu') & \text{by definition of } \theta_1 \end{array}$$

Note, summation in $\mathbb{B}$ is disjunction. Likewise, on the Markovian side, we have, for any $Q \in \mathcal{P}_{mal}$,

$$\begin{array}{lcll} \mathbf{M}(P, [Q]_R) & = & \sum_{Q' \in [Q]_R} \sum \{\!|\, \lambda \mid P \overset{\lambda}{\dashrightarrow} Q' \,|\!\} & \text{by definition of } \mathbf{M} \\ & = & \sum_{Q' \in [Q]_R} \mathscr{P}(Q') & \text{by Lemma 18b} \\ & = & \sum_{Q' \in [Q]_R} \theta_2(P)(\delta)(Q) & \text{by definition of } \theta_2 \end{array}$$

Comparing the equations following Definition 21 and the equations of Definition 23, we conclude that a strong bisimulation for *MAL* is also an $\mathcal{S}_{mal}$-bisimulation for the *FuTS* $\mathcal{S}_{mal}$, and vice versa. From this the theorem follows. □

As a corollary of the theorem we obtain that also for *MAL* the concrete notion of strong bisimilarity $\sim_{ma}$ is coalgebraically underpinned, as it coincides, with the behavioral equivalence $\simeq_{mal}$ that comes with the corresponding *FuTS* $\mathcal{S}_{mal}$.

11. 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 prominent 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 contributions of our paper include correspondence results, viz. Theorem 2, Theorem 7 and Theorem 15, that relate in the simple, combined and the new nested case, bisimilarity of a *FuTS* $\mathcal{S}$ to behavioural equivalence of the functor associated with $\mathcal{S}$. The result extends to general *FuTS* as well.

It is noted in [10], 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. [10, 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 [56], 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 [35], although no detailed proof is provided there, this will not happen with $\mathbb{R}_{\geqslant 0}$ as underlying semiring. In [25, Theorem 5.13] for $\mathcal{FS}(\cdot, \mathcal{M})$, with $\mathcal{M}$ a monoid, a characterization is given for weak preservation of pullbacks: $\mathcal{M}$ should be positive and refinable, i.e. (i) $m_1 + m_2 = 0$ iff $m_1, m_2 = 0$, and (ii) if $m_1 + m_2 = n_1 + n_2$ there exist p_{ij} such that $p_{i1} + p_{i2} = m_i$ and $p_{1j} + p_{2j} = n_j$ for $1 \leqslant i, j \leqslant 2$. The latter condition is also referred to as the row-column property for 2×2 matrices over $\mathcal{M}$, a property going back to [43]. In [39] 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 $i = 1 \ldots n$. The proof follows the set-up of [56], hence is different from [26]; we see that zero-sum coincides with positivity, while the existence of multiplicative inverses guarantees refinability. Thus, for semirings involved enjoying these properties, pullbacks are weakly preserved by $\mathcal{FS}(\cdot, \mathcal{R})$. Therefore, coalgebraic bisimilarity and behavioural equivalence are the same. As a consequence, under conditions which are met by the *SPC*s discussed in the preceding, we have that concrete bisimulation, *FuTS*-bisimilarity, behavioural equivalence and coalgebraic bisimilarity coincide.

For typical stochastic process languages based on *PEPA* and *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* as a stepping stone, the correspondence results bridge between the concrete notion of bisimulation for *PEPA* and *IML*, and the associated coalgebraic notions of behavioural equivalence. Hence, from this perspective, the concrete notions are seen as the natural strong equivalence to consider. Obviously, classical strong bisimilarity [42, 44] and bisimilarity for *FuTS* over $\mathbb{B}$ coincide (see [35] or [39] for details). Also, strong bisimulation of [31], an alternative to Hillston's notion of strong equivalence covered here, 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 of how languages based on discrete deterministic time, *TPC*, as well as those where stochastic time is integrated with discrete probability and with non-determinism, *MAL*, can be treated in the *FuTS* framework. A similar mediating role for *FuTS* applies to these calculi too: the concrete notion of bisimulation coincides with *FuTS* bisimulation, hence coincides with the corresponding behavioral equivalence.

As mentioned in Section 1, related work in the area of systematic approaches to frameworks for the semantics of *SPC*—and quantitative extensions of process calculi in general—includes the study of abstract quantitative GSOS, with its application to Weighted Transition Systems (*WTS*) [35, 34, 41]. Stochastic GSOS (SGSOS) and Weighted GSOS appear to be a special case of Miculan and Peressotti's *weight function* GSOS. In [35, 41] a treatment is given for *PEPA*, in line with Section 5 of the present paper. The formats mentioned above arise from the abstract theory of SOS. A noteworthy result, shown in [35], is that stochastic bisimilarity of *SPC* defined using the SGSOS format is guaranteed to be a congruence. The result is generalized to *WTS* in [34]. We did not address the issue of congruences for *FuTS* in the present paper. Nevertheless, we note that *Rated Transition Systems*—the semantic model used in [35]—are very similar to *RTS* of Latella, Massink

et al. [15, 17, 16], which are the instantiation of simple *FuTS* on non-negative real numbers, and that *WTS* are very similar to simple *FuTS*. Thus, it is to be expected that simple *FuTS* can be represented as *WTS* using the SGSOS, which would extend the congruence result to *FuTS*. The issue of the relationship with *WTS* remains, though, for the richer class of combined, nested, and general *FuTS*, which we leave for further study.

Acknowledgments The authors are grateful to Rocco De Nicola, Fabio Gadducci, Daniel Gebler, Michele Loreti, Jan Rutten, and Ana Sokolova for fruitful discussions on the subject and useful suggestions. The constructive comments by the reviewers have been of help and are much appreciated. DL and MM acknowledge support by EU Project n. 600708 *A Quantitative Approach to Management and Design of Collective and Adaptive Behaviours* (QUANTICOL). This research has been partially conducted while EV was spending a sabbatical leave at the CNR/ISTI. EV gratefully acknowledges the hospitality and support during his stay in Pisa.

References

[1] J. Adámek, S. Milius, and L.S. Moss. Initial algebras and terminal coalgebras: a survey. Preliminary version, 2010.
[2] J. Adámek and H.-E. Porst. On tree coalgebras and coalgebra presentations. *Theoretical Computer Science*, 311:257–283, 2004.
[3] A. Aldini, M. Bernardo, and F. Corradini. *A Process Algebraic Approach to Software Architecture design*. Springer, 2010.
[4] J.C.M. Baeten and C.A. Middelburg. *Process Algebra with Timing*. Springer, 2002.
[5] C. Baier, B.R. Haverkort, H. Hermanns, J.-P. Katoen, and M. Siegle, editors. *Validation of Stochastic Systems – A Guide to Current Research*. LNCS 2925, 2004.
[6] M. Bernardo. A survey of Markovian behavioral equivalences. In M. Bernardo and J. Hillston, editors, *SFM 2007 Advanced Lectures*, pages 180–219. LNCS 4486, 2007.
[7] M. Bernardo, R. De Nicola, and M. Loreti. A uniform framework for modeling nondeterministic, probabilistic, stochastic, or mixed processes and their behavioral equivalences. *Information and Computation*, 225(0):29 – 82, 2013.
[8] M. Bernardo and R. Gorrieri. A tutorial on EMPA: a theory of concurrent processes with non-determinism, priorities, probabilities and time. *Theoretical Computer Science*, 202(1–2):1–54, 1998.
[9] H.C. Bohnenkamp, P.R. D'Argenio, H. Hermanns, and J.-P. Katoen. MODEST: A compositional modeling formalism for hard and softly timed systems. *IEEE Transactions on Software Engineering*, 32(10):812–830, 2006.
[10] F. Bonchi, M. Bonsangue, M. Boreale, J. Rutten, and A. Silva. A coalgebraic perspective on linear weighted automata. *Information and Computation*, 211:77–105, 2012.
[11] M. Boreale. Weighted bisimulation in linear algebraic form. In M. Bravetti and G. Zavattaro, editors, *Proc. CONCUR 2009*, pages 163–177. LNCS 5710, 2009.
[12] M. Boreale and F. Gadducci. Processes as formal power series: A coinductive approach to denotational semantics. *Theoretical Computer Science*, 360(1–3):440–458, 2006.
[13] M. Bozga, A. David, A. Hartmanns, H. Hermanns, K.G. Larsen, A. Legay, and J. Tretmans. 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, 2012.
[14] J.W. de Bakker and E.P. de Vink. *Control Flow Semantics*. The MIT Press, 1996.
[15] R. De Nicola, D. Latella, M. Loreti, and M. Massink. Marcaspis: a markovian extension of a calculus for services. *Electr. Notes Theor. Comput. Sci.*, 229(4):11–26, 2009. Proceedings of SOS 2008, the 5th Workshop on Structural Operational Semantics, affiliated of ICALP 2008.
[16] R. De Nicola, D. Latella, M. Loreti, and M. Massink. On a uniform framework for the definition of stochastic process languages. In M. Alpuente, B. Cook, and C. Joubert, editors, *Formal Methods for Industrial Critical Systems, 14th International Workshop, FMICS 2009, Eindhoven, The Netherlands, November 2-3, 2009. Proceedings*, volume 5825 of *Lecture Notes in Computer Science*, pages 9–25. Springer, 2009.
[17] R. De Nicola, D. Latella, M. Loreti, and M. Massink. Rate-based transition systems for stochastic process calculi. In S. Albers et al., editor, *Proc. ICALP 2009, Part II*, pages 435–446. LNCS 5556, 2009.

[18] R. De Nicola, D. Latella, M. Loreti, and M. Massink. A Uniform Definition of Stochastic Process Calculi. *ACM Computing Surveys*, 46(1):5:1–5:35, 2013. DOI 10.1145/2522968.2522973.
[19] R. De Nicola, D. Latella, and M. Massink. Formal modeling and quantitative analysis of Klaim-based mobile systems. In H. Haddad et al., editor, *Proc. SAC 2005*, pages 428–435. ACM, 2005.
[20] Yuxin Deng and M. Hennessy. On the semantics of Markov automata. *Information and Computation*, 222:139–168, 2013.
[21] M. Droste, W. Kuich, and H. Vogler. *Handbook of Weighted Automata*. Monographs in Theoretical Computer Science. Springer, 2009.
[22] C. Eisentraut, H. Hermanns, and L. Zhang. Concurrency and composition in a stochastic world. In P. Gastin and F. Laroussinie, editors, *Proc. CONCUR 2010*, pages 21–39. LNCS 6269, 2010.
[23] C. Eisentraut, H. Hermanns, and Lijun Zhang. On probabilistic automata in continuous time. In *Proc. LICS, Edinburgh*, pages 342–351. IEEE Computer Society, 2010.
[24] R.J. van Glabbeek, S.A. Smolka, and B. Steffen. Reactive, generative and stratified models of probabilistic processes. *Information and Computation*, 121(1):59–80, 1995.
[25] H.P. Gumm and T. Schröder. Monoid-labeled transition systems. *Electronic Notes in Theoretical Computer Science*, 44(1):185–204, 2001.
[26] H.P. Gumm and T. Schröder. Products of coalgebras. *Algebra Universalis*, 46:163–185, 2001.
[27] H. Hermanns. *Interactive Markov Chains*. LNCS 2428, 2002.
[28] H. Hermanns, U. Herzog, and J.-P. Katoen. Process algebra for performance evaluation. *Theoretical Computer Science*, 274(1–2):43–87, 2002.
[29] H. Hermanns, U. Herzog, and V. Mertsiotakis. Stochastic process algebras – between LOTOS and Markov chains. *Computer Networks and ISDN Systems*, 30:901–924, 1998.
[30] H. Hermanns and J.-P. Katoen. The how and why of Interactive Markov Chains. In F.S. de Boer, M.M. Bonsangue, S. Hallerstede, and M. Leuschel, editors, *Proc. FMCO 2009*, pages 311–337. LNCS 6286, 2010.
[31] J. Hillston. *A Compositional Approach to Performance Modelling*, volume 12 of *Distinguished Dissertations in Computer Science*. Cambridge University Press, 1996.
[32] J. Hillston. Process algebras for quantitative analysis. In *Proc. LICS, Chicago*, pages 239–248. IEEE, 2005.
[33] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
[34] B. Klin. Structural operational semantics for weighted transition systems. In J. Palsberg, editor, *Semantics and Algebraic Specification*, pages 121–139. LNCS 5700, 2009.
[35] B. Klin and V. Sassone. Structural operational semantics for stochastic process calculi. In R.M. Amadio, editor, *Proc. FoSSaCS 2008*, pages 428–442. LNCS 4962, 2008.
[36] J.N. Kok and J.J.M.M. Rutten. Contractions in comparing concurrency semantics. *Theoretical Computer Science*, 76:179–222, 1990.
[37] A. Kurz. *Logics for coalgebras and applications to computer science*. PhD thesis, LMU München, 2000.
[38] D. Latella, M. Massink, and E.P. de Vink. Bisimulation of labeled state-to-function transition systems of stochastic process languages. In T. Soboll and U. Golas, editors, *Proc. ACCAT 2012, Tallin*, pages 23–43. EPTCS 93, 2012.
[39] D. Latella, M. Massink, and E.P. de Vink. Coalgebraic Bisimulation of FuTS. Technical Report 09, ASCENS–Autonomic Service-Component Ensembles (EU Project 257414), 2013.
[40] D. Latella, M. Massink, and E.P. de Vink. A definition scheme for quantitative bisimulation. In N. Bertrand and M. Tribastone, editors, *Proc. QAPL 2015*, pages 63–78. EPTCS 194, 2015.
[41] M. Miculan and M. Peressotti. GSOS for non-deterministic processes with quantitative aspects. In N. Bertrand and L. Bortolussi, editors, *Proc. QAPL 2014*, pages 17–33. EPTCS 154, 2014.
[42] R. Milner. *A Calculus of Communicating Systems*. LNCS 92, 1980.
[43] L. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96:277–317, 1999.
[44] D. Park. Concurrency and automata on infinite sequences. In *Proc. GI-Conference 1981, Karlsruhe*, pages 167–183. LNCS 104, 1981.
[45] C. Priami. Stochastic π-calculus. *The Computer Journal*, 38(7):578–589, 1995.
[46] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249:3–80, 2000.
[47] J.J.M.M. Rutten. Behavioural differential equations: a coinductive calculus of streams, automata, and power series. *Theoretical Computer Science*, 308(1–3):1–53, 2003.
[48] A. Silva. *Kleene Coalgebra*. PhD thesis, Radboud University Nijmegen, 2010.
[49] A. Silva, F. Bonchi, M. Bonsangue, and J. Rutten. Quantitative Kleene coalgebras. *Information and Computation*, 209(5):822–846, 2011.

[50] A. Sokolova. Probabilistic systems coalgebraically: a survey. *Theoretical Computer Science*, 412(38):5095–5110, 2011.
[51] S. Staton. Relating coalgebraic notions of bisimulation. *Logical Methods in Computer Science*, 7:1–21, 2011.
[52] M. Timmer. *Efficient Modelling, Generation and Analysis of Markov Automata*. PhD thesis, University of Twente, 2013.
[53] M. Timmer, J.-P. Katoen, J. van de Pol, and M. Stoelinga. Efficient modelling and generation of Markov automata. In M. Koutny and I. Ulidowski, editors, *Proc. CONCUR 2012*, pages 364–379. LNCS 7454, 2012.
[54] M. Timmer, J.-P. Katoen, J. van de Pol, and M. Stoelinga. Efficient modelling and generation of Markov automata (extended version). Technical Report TR-CTIT 12-16, CTIT, Universiteit Twente, 2012.
[55] D. Turi and G.D. Plotkin. Towards a mathematical operational semantics. In *Proc. LICS 1997, Warsaw*, pages 280–291. IEEE, 1997.
[56] E.P. de Vink and J.J.M.M. Rutten. Bisimulation for probabilistic transition systems: a coalgebraic approach. *Theoretical Computer Science*, 221:271–293, 1999.
[57] U. Wolter. CSP, partial automata, and coalgebras. *Theoretical Computer Science*, 280:3–34, 2002.