



Reduction Semantics in Markovian Process Algebra

Mario Bravetti

► To cite this version:

Mario Bravetti. Reduction Semantics in Markovian Process Algebra. Journal of Logical and Algebraic Methods in Programming, 2018, 96, pp.41-64. 10.1016/j.jlamp.2018.01.002 . hal-01921194

HAL Id: hal-01921194

<https://inria.hal.science/hal-01921194>

Submitted on 13 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Reduction Semantics in Markovian Process Algebra

Mario Bravetti

*University of Bologna, Department of Computer Science and Engineering / FOCUS INRIA
Mura Anteo Zamboni 7, 40126 Bologna, Italy*

Abstract

Markovian process algebras allow for performance analysis by automatic generation of Continuous Time Markov Chains. The inclusion of exponential distribution rate information in process algebra terms, however, causes non trivial issues to arise in the definition of their semantics. As a consequence, technical settings previously considered do not make it possible to base Markovian semantics on directly computing reductions between communicating processes: this would require the ability to readjust processes, i.e. a commutative and associative parallel operator and a congruence relation on terms enacting such properties. Semantics in reduction style is, however, often used for complex languages, due to its simplicity and conciseness. In this paper we introduce a new technique based on stochastic binders that allows us to define Markovian semantics in the presence of such a structural congruence. As application examples, we define the reduction semantics of Markovian versions of the π -calculus, considering both the cases of Markovian durations: being additional standalone prefixes (as in Interactive Markov Chains) and being, instead, associated to standard synchronizable actions, giving them a duration (as in Stochastic π -calculus). Notably, in the latter case, we obtain a “natural” Markovian semantics for π -calculus (CCS) parallel that preserves, for the first time, its associativity. In both cases we show our technique for defining reduction semantics to be correct with respect to the standard Markovian one (in labeled style) by providing Markovian extensions of the classical π -calculus Harmony theorem.

Keywords: Process Algebra, Structural Operational Semantics, Markov Chains

1. Introduction

The importance of considering probabilistic and timing aspects in system specification and analysis is widely recognized: first of all, the behavior of distributed systems and communication protocols often depends on probability and time; second, expressing duration of system activities makes it possible to estimate system performance. Markovian process algebras (see, e.g., [17, 1, 21, 16, 7, 4]) extend classical process algebras with probabilistic exponentially distributed time durations denoted by rates λ (positive real numbers), where λ is the parameter of an exponential distribution. Defining an operational

semantics for a Markovian process algebra makes it possible to derive Continuous Time Markov Chains from system specifications, which can then be analyzed for performance. Essentially this requires generating, from rates λ expressed syntactically in the prefixes of the process algebra, transitions labeled by rates.

Markovian process algebras in the literature have been defined in technical settings where it is not possible to compute their transitions by readjusting parallel processes, i.e. to represent their evolution by (not action labeled) *reduction* transitions (see, e.g., the reduction semantics of π -calculus [22, 19]). This is due to limitations in existing approaches to Markovian operational semantics that make them not compatible with a *structural congruence relation*. In reduction semantics such a relation is endowed with commutative and associative laws of parallel that allow communicating processes to get syntactically adjacent to each other in order to directly produce reduction transitions. The incompatibility arises from the techniques used for managing *multiplicity of rate transitions*: a Markovian semantics, in order to be correct must express multiplicity of several identical transitions, that is transitions with the same rate λ and source and target terms. Process algebras enriching prefixes with Markovian rates λ previously introduced in the literature use, instead, semantical definitions in the *labeled operational semantics* style: transitions are labeled with actions (representing potential of communication) and actual communications are produced by matching transitions of parallel processes according to their action labels. Reduction semantics is, however, widely used, since, due to its simplicity and conciseness, it is convenient for defining semantics of complex languages, see, e.g., [18].

Moreover, for Markovian process algebras where CCS/ π -calculus parallel is used and both output and input actions are quantified by rates/weights, existing approaches in the literature (e.g. stochastic π -calculus of [21]) do not even guarantee associativity of parallel. That is in general, unless specific restrictions on the structure of terms are assumed (see [9]), different rates are obtained for outgoing transitions of terms $(P|Q)|R$ and $P|(Q|R)$.

In this paper we present a technique that allows us, for the first time, to develop semantics for Markovian process algebra in reduction style, i.e., where transitions are not labeled with actions and are directly produced by readjusting the structure of terms via a structural congruence relation. This opens the new possibility to include rates λ in actions of complex languages like that of [18] with no need to completely change the semantics from reduction to labeled style and without losing important properties like associativity of parallel. Notably, for the above mentioned case of Markovian process algebras where both output and input actions are quantified by rates/weights, our reduction semantics is the first Markovian one preserving *associativity of CCS/ π -calculus parallel*.

More precisely, we consider stochastic versions of the π -calculus, encompassing both the case of rates being additional standalone prefixes and rates being, instead, attached to standard action prefixes. In the former case, *Markovian delay* prefixes (λ) are expressed separately from standard actions (that are unmodified) as in the Interactive Markov Chains approach of [16]. In the latter case, we instead decorate π -calculus output and τ prefixes with a λ rate, denoting *Markovian actions*. We consider two variants for this case. In the first variant

input prefixes are not explicitly quantified (they are syntactically unmodified) and τ resulting from the synchronization of a λ decorated output on channel “ a ” and an input on “ a ” simply has rate λ : this corresponds to multiplying λ by the number of possible synchronizations over “ a ”, similarly to what is done in the context of biochemistry, see e.g. [8]. In the second variant input prefixes are, instead, endowed with weights w , i.e. numbers establishing the probability to choose an input on channel “ a ” given that “ a ” is the channel of the selected output: this corresponds to subdividing the output rate λ among the possible “ a ” synchronizing inputs according to their weights, as done in Stochastic π -calculus [21] by taking inputs to have (weight equipped) unspecified rates [17, 21].

In all above cases, we first define the semantics of the Markovian calculus in labeled operational semantics style (we use the early semantics of the π -calculus) using the classical approach of [14, 21] to deal with multiplicity of identical transitions. We then apply our technique, which, as we will see, is based on the use of so-called *stochastic names* and *stochastic binders* (for both rates and weights) to provide a reduction semantics for the same process algebra.

We finally show our approach to correctly calculate rates (manage their multiplicity) by means of a theorem similar to the Harmony Lemma in [22, 19], that is: stochastic reductions of reduction semantics are in correspondence with Markovian (τ) transitions of the labeled operational semantics and, thus, the underlying Markov chains. We show such a theorem to hold both for Markovian delays and actions, but not for the variant with input weights: this is related to the fact that parallel of Stochastic π -calculus [21] is *not* associative. If, e.g., in [21] we consider $P|(P'|P'')$ with $P = \bar{x}\langle y \rangle_3.Q$, $P' = x(z)_4.Q'$ and $P'' = x(z')_8.Q''$ (we represent a prefix rate as a subscript), we get two distinguished τ transitions with rates 1 and 2. That is, rate 3 of the output multiplied by the probability to choose, inside $P'|P''$ (the subterm to the right of the parallel where synchronization happens): either input $x(z)_4$, i.e. $4/12$ (the rate of the input divided by the total rate of inputs on x in $P'|P''$); or input $x(z')_8$ in $P'|P''$, i.e. $8/12$ (similarly calculated). Rates of inputs are therefore treated as weights, according to which the output rate 3 is distributed. If, instead, we consider $(P|P')|P''$ [21] yields two distinguished τ transitions with the same rate 3. That is, rate 3 of the output multiplied by the probability: to choose $x(z)_4$ inside P' , i.e. $4/4$ (the rate of the input divided by the total rate of inputs on x in P'); and to choose input $x(z')_8$ inside P'' , i.e. $8/8$.¹

As shown also in [12] non associativity of parallel comes from the fact that [21] calculates rates, upon synchronization, in a way that strictly depends on a fixed structure for parallel operators in the term (independently of the particular labeled semantic technique used to deal with rates - [9], [12] or the original [21]):

¹In this example rates of inputs are treated as weights because, for all parallel operators $P_1|P_2$ considered, the total rate of outputs on x in P_1 is smaller than the total rate of input on x in P_2 (see [21]): the same is obtained by taking rates of inputs as (weight equipped) unspecified rates of [17, 21].

the rate of a synchronization on channel x performable by $P_1 | P_2$, depends on the total rate of input/output transitions on x performable by P_1 and P_2 , called *apparent rate* in [17, 21]. By using our reduction semantics technique, instead, the calculation of the total weight is not based on such a fixed structure, but it is the sum of weights of *all* synchronizable input actions (independently of how parallel is associated), hence *parallel is dealt with in an associative way*: in the example above we consider for the calculation of the total weight all input actions synchronizable with $\bar{x} < y >_3$, thus getting two reductions with rates 1 and 2 independently on how parallel is associated. This is natural in the context of π -calculus (or CCS) parallel $P_1 | P_2$ in which, differently from CSP parallel (originally used when Markovian semantics and apparent rate was defined [17]), an action of P_1 that can be synchronized with one of P_2 can also be synchronized with one performed by an outer parallel process.

1.1. The Problem of Dealing with Structural Congruence

We now explain why, due to the technique used for expressing multiplicity of identical transitions, previous approaches for extending process algebras with Markovian durations are not compatible with the use of a structural congruence, as needed in reduction semantics.

Among techniques previously considered, the simplest one (see [17]) is counting the number of different possible inferences of the same transition in order to determine its multiplicity. However this can be done only for semantic definitions that do not generate transitions by means of a structural congruence relation \equiv on terms, that is via the classical operational rule of closure w.r.t. \equiv that makes it possible to infer a transition $P \xrightarrow{\alpha} Q'$ from $Q \xrightarrow{\alpha} Q'$ whenever $P \equiv Q$. This is because, the presence of such a rule causes the same transition to be inferred in a number of ways even in the case of just one occurrence of a Markovian delay; e.g., considering P just being a single Markovian delay and the common congruence law $P \equiv P|0$, such a number would even be infinite.

Another similar way used in the literature (see, e.g., [14, 21]) to deal with Markovian transition multiplicity is to explicitly introduce a mechanism to differently identify multiple executions of identical Markovian delays by introducing some kind of identifier that typically expresses syntactic position of the delay in the term (e.g. it is called a “location” in the case of the position with respect to the parallel operator) and whose syntax is *dependent* on the operators used in the language. Again, since Markovian transitions are, roughly, identified by indicating the path (left/right at every node) in the syntax tree of the term, this solution is not compatible with using a structural congruence relation with commutative and associative laws that readjusts term structure at need during inference. As already said, we will use this approach to express the labeled operational semantics of the Markovian process algebras that we consider in this paper and for comparison with our reduction semantics.

Finally, techniques exploiting the representation of Markovian transitions collectively as a distribution over a target state space (a traditional approach in the context of probabilistic transition systems, see Giry functor [13] and its generalizations, e.g., [23, 11, 24]) have been used in [9] and in [12] for any

Markovian process algebra whose operational rules satisfy a Markovian extension of the GSOS rule format. These techniques are crucially based on *collectively* inferring Markovian transitions performable at a certain syntactic level (e.g. in $P|Q$) from transitions performable at lower levels (e.g. by P and by Q). Therefore, they are, again, not compatible with using structural congruence \equiv (with commutative and associative laws) for generating transitions, as caused by the operational rule of closure w.r.t. \equiv mentioned above. Formally, such a rule is indeed not in the GSOS format of [12], hence the approach of [12] cannot be used in its presence. It is, however, a fundamental rule in reduction semantics. Conceptually, the reason for incompatibility is that, due to this rule, different transitions may need different structural readjustments to be inferred: e.g. given term $P|(Q|R)$ it can be that it needs to be readjusted into $(P|Q)|R$ to infer a Markovian reduction due to P communicating with Q and to $(P|R)|Q$ to infer a Markovian reduction due to P communicating with R . So we cannot fix a stratification of transition inference, that is first *collectively* inferring all transitions of $(Q|R)$ and then use them to infer all transitions of $P|(Q|R)$: for each Markovian reduction of $P|(Q|R)$ it must be possible to *individually* perform a different readjustment of parallel.

1.2. Main Idea

In this paper we present a technique for defining semantics of a Markovian process algebra in the presence of structural congruence. This allows us to define Markovian semantics in reduction style. As already mentioned, our technique is applicable for Markovian extensions of process algebra obtained: either by simply adding Markovian (λ) delay prefixes or by decorating existing action prefixes with λ . Our technique is also compatible with standard unfolding of recursion made by structural congruence: the possibility of expressive cycling behavior via a recursion operator is fundamental for steady state based performance analysis. Moreover, our technique is such that the inclusion of rates λ in process algebra prefixes does not cause a significant modification of process algebra semantics, in that most of the burden is concentrated in few additional rules which are independent of the particular operators used by the algebra. More precisely, we do not have to introduce an explicit (operator dependent) mechanism to distinguish multiple occurrences of actions as in [14, 21], or to intervene in the semantics of every operator as in [9] or to separately define, by induction on term syntax, how rate distributions are calculated using (total) rate information attached to rules as in [12] (which also does not cope with recursion).

The idea is to introduce *stochastic names* (that can be seen as names for stochastic variables) and *stochastic binders* (binders for such names). Stochastic names are used in process algebra prefixes and rates λ are assigned to them by the corresponding stochastic binder. In essence (a set of) stochastic binders act as a (probabilistic/stochastic) scheduler: they use names attached to prefixes to collectively quantify them. In this way, *assuming that all prefixes are associated with a different stochastic name*, rate values replace stochastic names upon binding and multiplicity is correctly taken into account because prefixes with the same rates are anyhow distinguished by having different stochastic names.

But how to guarantee that prefixes are given different stochastic names? The idea is to consider a specification language where rates λ are in prefixes and then to turn them (by laws of structural congruence) into stochastic names that are immediately bound by a stochastic binder assigning the rate λ . Then, with the usual rules of binder extrusion, we impose that, before binders can be evaluated, they must be all lifted to the outermost syntactic level by structural congruence. As a consequence we have the guarantee (due to the requirement of α -conversion related to binder extrusion) that every prefix is assigned a distinguished stochastic name. Note that recursion is dealt with in the correct way as well, because, as we will see, recursions need to be unfolded in order to perform the lift of the binders.

1.3. Paper Outline

The paper is structured as follows. Section 2 presents the extension of π -calculus with Markovian delays (λ) and its labeled operational semantics according to the classical technique of [21]. Section 3 introduces our technique, uses it for defining semantics in reduction style of the algebra of Section 2 and shows the obtained semantics to be correct w.r.t. the labeled one via an Harmony theorem. Section 4 considers the π -calculus extension with Markovian actions in both versions (with and without weights) and, for each of them, presents: its labeled semantics according to the classical technique of [21], our semantics in reduction style and comparison results via a second Harmony theorem. Finally, Section 5 concludes the paper by further discussing related work. Appendices A and B include proofs of the more involved results presented in Section 3 and 4, respectively. Initial work was carried out in the technical report [2].

2. Classical Approach for Markovian Delays

We consider the π -calculus with constant definitions (as in [20]) extended with Markovian delay prefixes, following the Interactive Markov Chain approach of [16]. Markovian delays are thus simply represented by (λ) prefixes, where λ is the rate of an exponential distribution. As usual in the context of Markovian process algebra, see, e.g., [21]: we consider recursion by constant definitions (as done for the standard π -calculus, e.g., in [20]) and not by a bang operator; and we assume recursion to be (weakly) guarded. In this way we avoid terms like $A \triangleq A|(\lambda).P$ for which we would not get a finite cumulative rate for the outgoing Markovian transitions of A .

Let x, y, \dots range over the set \mathcal{X} of channel names and λ range over the set \mathbf{R}^+ of rate values, representing rates of exponential distributions.

Definition 2.1. We take \mathcal{P} to be the set of terms P, Q, \dots generated by

$$\begin{aligned} P &::= M \mid P|P' \mid (\nu x)P \mid A(x_1, \dots, x_n) \\ M &::= \underline{0} \mid \alpha.P \mid M + M' \\ \alpha &::= \bar{x} < y > \mid x(y) \mid \tau \mid (\lambda) \end{aligned}$$

We abbreviate x_1, \dots, x_n with \tilde{x} and we assume a given set of constant definitions $A(\tilde{x}) \triangleq P$ such that the following is satisfied. We can, by starting from P and by performing a finite number of successive substitutions of constants $B(\tilde{y})$ for their defining term $Q\{\tilde{y}/\tilde{x}\}$ (assuming $B(\tilde{x}) \triangleq Q$), obtain a term P' where every constant occurs in P' in the scope of a prefix operator (A is weakly guarded defined, see [20]).

Concerning the labeled operational semantics we consider the early semantics of the π -calculus (see, e.g., [22]) and, concerning Markovian delays, the classical approach of [21], i.e. we use an explicit (operator dependent) mechanism to distinguish multiple occurrences.

Operational semantics is thus defined by the following labeled transition system. We assume η to range over its labels and transitions be of two kinds:

- The classical π -calculus early semantics labeled transitions, defined in Table 1. We assume: symmetric rules to be also included, as usual; the meaning of labels and the definition of functions n (names), bn (bound names) and fn (free names) to be the standard one, see, e.g., [22]; and μ to range over the set of standard labels, i.e. $\bar{x}y$, $\bar{x}(y)$, xy or τ labels.
- Markovian transitions, defined via the additional rules in Table 2, that are labeled with pairs “ λ, id ” where: λ is the rate of the Markovian delay being executed and id is a unique identifier. An identifier id is a string over the alphabet $\{+0, +1, |0, |1\}$ ending with an additional “ \bullet ” and represents the syntactic position of the delay in the syntax tree of the term (once, constant replacement is applied if needed), i.e. the path in the tree.

Notice that η ranges over both standard labels μ and labels “ λ, id ” of Markovian transitions, which are defined by the rules in Table 2 and the relevant rules of Table 1.

Example 2.1. Consider $P = (3).P' + (3).P'$. We have that P performs transitions:

$$\begin{aligned} P &\xrightarrow{3, +0\bullet} P' \\ P &\xrightarrow{3, +1\bullet} P' \end{aligned}$$

Notice that, thanks to the use of identifiers, we get two transitions for P (without them we would have got just one) and we correctly have that the total rate of transiting from P to P' is 6.

Consider now $P|Q$, with P as above. It performs (P originated) transitions:

$$\begin{aligned} P|Q &\xrightarrow{3, |0 +0\bullet} P'|Q \\ P|Q &\xrightarrow{3, |0 +1\bullet} P'|Q \end{aligned}$$

Again, thanks to the use of identifiers, we correctly have that the obtained overall rate of (P originated) transitions from $P|Q$ to $P'|Q$ is 6.

$$\begin{array}{c}
\tau.P \xrightarrow{\tau} P \quad \bar{x}\langle y \rangle.P \xrightarrow{\bar{x}y} P \quad x(y).P \xrightarrow{xy'} P\{y'/y\} \\
\\
\frac{P \xrightarrow{\mu} P'}{P+Q \xrightarrow{\mu} P'} \quad \frac{P \xrightarrow{\mu} P'}{P|Q \xrightarrow{\mu} P'|Q} \quad \text{bn}(\mu) \cap \text{fn}(Q) = \emptyset \\
\frac{P \xrightarrow{\bar{x}y} P' \quad Q \xrightarrow{xy} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \quad \frac{P \xrightarrow{\bar{x}(y)} P' \quad Q \xrightarrow{xy} Q'}{P|Q \xrightarrow{\tau} (\nu y)(P'|Q')} \quad y \notin \text{fn}(Q) \\
\\
\frac{P \xrightarrow{\eta} P'}{(\nu x)P \xrightarrow{\eta} (\nu x)P'} \quad x \notin n(\eta) \quad \frac{P \xrightarrow{\bar{x}y} P'}{(\nu y)P \xrightarrow{\bar{x}(y)} P'} \quad y \neq x \\
\\
\frac{P\{\tilde{y}/\tilde{x}\} \xrightarrow{\eta} P'}{A(\tilde{y}) \xrightarrow{\eta} P'} \quad A(\tilde{x}) \triangleq P
\end{array}$$

Table 1: Standard Rules (Symmetric Rules Omitted)

$$\begin{array}{c}
(\lambda).P \xrightarrow{\lambda, \bullet} P \\
\\
\frac{P \xrightarrow{\lambda, id} P'}{P+Q \xrightarrow{\lambda, +_0 id} P'} \quad \frac{Q \xrightarrow{\lambda, id} Q'}{P+Q \xrightarrow{\lambda, +_1 id} Q'} \\
\\
\frac{P \xrightarrow{\lambda, id} P'}{P|Q \xrightarrow{\lambda, |_0 id} P'|Q} \quad \frac{Q \xrightarrow{\lambda, id} Q'}{P|Q \xrightarrow{\lambda, |_1 id} P|Q'}
\end{array}$$

Table 2: Identified Markovian Transitions

The labeled transition system arising from the semantics of a term P has states with both outgoing Markovian and standard transitions: it has to be interpreted as an Interactive Markov Chain of [16], with τ transitions being instantaneously executed and other π -calculus labeled transitions representing potential for execution. In particular Markovian transitions performable in states that also have outgoing τ transitions are considered to be pre-empted (so-called maximal progress assumption) and a system is considered to be complete if it cannot undergo further communication with the environment, i.e. it can perform only τ and Markovian transitions.

This is expressed by the notion of Markovian Bisimulation considered in [16] and that we also report here. We first define the total rate $\gamma(P, P')$ of transiting from state P to P' via Markovian transitions as

$$\gamma(P, P') = \sum_{P \xrightarrow{\lambda, id} P'} \lambda$$

For instance in the above Example 2.1 we have $\gamma(P, P'|Q) = 6$. This is extended

to sets of states in the usual way:

$$\gamma(P, Set) = \sum_{P' \in Set} \gamma(P, P')$$

We can now consider the Markovian bisimulation as in [16] (see [16] for weak bisimulation definition abstracting, when possible, from τ transitions).

Definition 2.2. *An equivalence relation \mathcal{B} on \mathcal{P} is a strong bisimulation iff $P \mathcal{B} Q$ implies for all μ and all equivalence classes C of \mathcal{B}*

- $P \xrightarrow{\mu} P'$ implies $Q \xrightarrow{\mu} Q'$ for some Q' with $P' \mathcal{B} Q'$
- $P \not\xrightarrow{\tau}$ implies $\gamma(P, C) = \gamma(Q, C)$

Two processes P and Q are strongly bisimilar if (P, Q) is contained in some strong bisimulation \mathcal{B} .

3. Reduction Semantics for Markovian Delays

We now use our technique based on symbolic representation of rates via stochastic names and binders to give a semantics in *reduction style* for the Markovian π -calculus of Section 2.

3.1. Stochastic Names and Stochastic Binders

We extend the syntax of terms P, Q, \dots by *adding* stochastic names and stochastic binders. We use $q, q', \dots \in \mathcal{QN}$ to range over stochastic names, i.e. names used to express symbolically stochastic information for delays. Moreover θ, θ', \dots ranges over channel names and stochastic names, i.e. the set $\mathcal{X} \cup \mathcal{QN}$.

Definition 3.1. *We take \mathcal{P}_{ext} to be the set of terms P, Q, \dots generated by extending the syntax of \mathcal{P} terms in Definition 2.1 as follows:*

$$\begin{aligned} P &::= \dots \mid (\nu q \rightarrow \lambda)P \\ \alpha &::= \dots \mid (q) \end{aligned}$$

The idea is that we can now represent (λ) delays by means of a stochastic name q that can be seen as the name of a stochastic variable. More precisely a delay (λ) can be equivalently represented by (q) in the scope of a stochastic binder $(\nu q \rightarrow \lambda)$ that associates rate λ to the stochastic name q (i.e. the binder “quantifies” q). In the following we use $(\nu x \rightarrow \varepsilon)$ to stand for (νx) , i.e. the standard π -calculus binder, which does not associate any “quantification” to x . This will allow us to write $(\nu \theta \rightarrow \widehat{\lambda})$ to stand for any binder (stochastic or classical) by assuming $\widehat{\lambda}, \widehat{\lambda}'$ to range over $\mathbf{R}^+ \cup \{\varepsilon\}$.

Notice that we will use \mathcal{P}_{ext} terms over the extended syntax to define the semantics of \mathcal{P} terms in reduction style. As we will see, for every state of the semantics we will always have a congruent term which belongs to \mathcal{P} . The definition of structural congruence follows.

$(\nu\theta \rightarrow \hat{\lambda})P \mid Q$	\equiv	$(\nu\theta \rightarrow \hat{\lambda})(P \mid Q)$	if $\theta \notin \text{fn}(Q)$
$(\nu\theta \rightarrow \hat{\lambda})(\nu\theta' \rightarrow \hat{\lambda}')P$	\equiv	$(\nu\theta' \rightarrow \hat{\lambda}')(\nu\theta \rightarrow \hat{\lambda})P$	if $\theta \neq \theta'$
$(\nu\theta \rightarrow \hat{\lambda})\underline{0}$	\equiv	$\underline{0}$	
$(M_1 + M_2) + M_3$	\equiv	$M_1 + (M_2 + M_3)$	
$M + N$	\equiv	$N + M$	
$M + \underline{0}$	\equiv	M	
$(P_1 \mid P_2) \mid P_3$	\equiv	$P_1 \mid (P_2 \mid P_3)$	
$P \mid Q$	\equiv	$Q \mid P$	
$P \mid \underline{0}$	\equiv	P	
$A(\tilde{y})$	\equiv	$P\{\tilde{y}/\tilde{x}\}$	if $A(\tilde{x}) \triangleq P$

Table 3: Standard Structural Congruence Laws (with quantified binders)

We consider a structural congruence relation over processes in \mathcal{P}_{ext} defined as usual by a set of laws. We consider the standard laws in Table 3 (where we just add the possible presence of quantification in binders) and, in addition, the following law:

$$(\lambda).P + M \equiv (\nu q \rightarrow \lambda)((q).P + M) \quad \text{if } q \notin \text{fn}(M, P)$$

As usual, *alpha renaming* inside processes, for any name θ and its binder $(\nu\theta \rightarrow \hat{\lambda})$ is assumed. Concerning the use of parentheses when writing terms in the laws above and in the following, we assume binding to take precedence over parallel and prefix to take precedence over sum.

3.2. Operational Semantics

We now define the semantics of the process algebra of Definition 3.1 in reduction style. As we will see besides standard reduction transitions (defined with standard rules), we have two other kinds of transitions representing stochastic execution. Transitions labeled with a stochastic (delay) name, which are treated similarly to standard reduction transitions, and stochastic transitions, which are generated when a stochastic binder is applied at the top level. The latter lead from a term to a *rate distribution* over terms.

We consider rate distributions σ , in the style of the probabilistic model of [23] (also used by [9] in the context of Markovian rates), describing how a state can be reached from another one by performing a Markovian transition.

Definition 3.2. *A rate distribution σ over a countable set S is a non-empty finite multiset over $\mathbf{R}^+ \times S$, i.e. a multiset of pairs (λ, e) , with $\lambda \in \mathbf{R}^+$ and $e \in S$. We use $RDist(S)$ to denote the set of rate distributions over S .*

Due to use of the structural congruence relation, it will be convenient to use rate distributions σ over congruence classes of terms, i.e. $\sigma \in RDist(\mathcal{P}_{ext}/\equiv)$, instead of single terms (as, e.g., done in [9]).

The operational semantics is defined in terms of three transition relations:

$$\begin{array}{c}
(q).P + M \xrightarrow{q} P \quad \tau.P + M \longrightarrow P \\
\bar{x}\langle y \rangle.P + M \mid x(z).Q + N \longrightarrow P \mid Q\{y/z\} \\
\frac{P \xrightarrow{\hat{q}} P'}{P|Q \xrightarrow{\hat{q}} P'|Q} \quad \frac{P \xrightarrow{\hat{q}} P'}{(\nu\theta \rightarrow \hat{\lambda})P \xrightarrow{\hat{q}} (\nu\theta \rightarrow \hat{\lambda})P'} \quad \theta \neq \hat{q} \\
\frac{P \equiv Q \quad Q \xrightarrow{\hat{q}} Q' \quad Q' \equiv P'}{P \xrightarrow{\hat{q}} P'}
\end{array}$$

Table 4: Standard Rules for Reductions and Symbolically Quantified Transitions

- Standard reductions [19], denoted by $P \longrightarrow P'$, that represent action execution.
- Transitions similar to the standard reductions of [19] that represent delay executions and just differ from reductions in that they are additionally labeled with the name q associated with the delay: their quantification is symbolic in that the name of the delay is included and not the actual rate. They are denoted by $P \xrightarrow{q} P'$.
- Stochastic transitions leading from a term to a rate distribution over congruence classes of terms. They are denoted by $P \longrightarrow \sigma$, where $\sigma \in RDist(\mathcal{P}_{ext}/\equiv)$.

The first two relations are collectively defined as the smallest subset of $\mathcal{P}_{ext} \times (\mathcal{QN} \cup \{\varepsilon\}) \times \mathcal{P}_{ext}$ (where ε labeled reductions represent the standard ones of [19], i.e. $\xrightarrow{\varepsilon}$ stands for \longrightarrow) satisfying the operational rules in Table 4. We assume \hat{q} , used in Table 4 to denote transition labels, to range over stochastic names q and ε , i.e. $\hat{q} \in \mathcal{QN} \cup \{\varepsilon\}$. Notice that, with respect to the standard ones of [19], the rules in Table 4 only differ in including a rule for (q) prefix and in considering names $q \in \mathcal{QN}$ as possible reduction labels (and rates λ possibly associated to binders).

Stochastic $P \longrightarrow \sigma$ transitions are defined as the smallest subset of $\mathcal{P}_{ext} \times RDist(\mathcal{P}_{ext}/\equiv)$ satisfying the operational rules in Table 5, where we assume $(\nu q \rightarrow \lambda)\sigma = \{\} (\lambda', [(\nu q \rightarrow \lambda)P]_{\equiv}) \mid (\lambda', [P]_{\equiv}) \in \sigma \}$.² Notice that, even if *(Sto2)* rule includes a negative premise, the operational semantics is well-defined in that the inference of transitions can be stratified (see, e.g., [15]).

The idea behind the semantics is that, since in Table 5 we do not have rules for other operators besides stochastic binders, in order for a stochastic $P \longrightarrow \sigma$ transition to be generated, term P must be rearranged by structural congruence in such a way that all its stochastic binders (those actually binding a (q) delay prefix) are the outermost operators, i.e. they must be all extruded up to the top

²We use $\{\}$ and $\{\}$ to denote multiset parentheses and \cup to denote multiset union.

$$\begin{array}{c}
(Sto1) \frac{P \longrightarrow \sigma \quad P \xrightarrow{q} P'}{(\nu q \rightarrow \lambda)P \longrightarrow (\nu q \rightarrow \lambda)\sigma \cup \{(\lambda, [(\nu q \rightarrow \lambda)P']_{\equiv})\}} \\
(Sto2) \frac{\nexists \sigma : P \longrightarrow \sigma \quad P \xrightarrow{q} P'}{(\nu q \rightarrow \lambda)P \longrightarrow \{(\lambda, [(\nu q \rightarrow \lambda)P']_{\equiv})\}} \\
(StoCong) \frac{P \equiv Q \quad Q \longrightarrow \sigma}{P \longrightarrow \sigma}
\end{array}$$

Table 5: Rules for Stochastic Transitions

level. For instance, considering $P = (\nu q \rightarrow \lambda)((q) | (\nu q' \rightarrow \lambda')(q'))$, one could think that a stochastic transition for P , erroneously containing the λ rate only, could be inferred by the rules, i.e. a “partial” stochastic transition. On the contrary, this cannot happen because $(q) | (\nu q' \rightarrow \lambda')(q')$ does *not* satisfy the negative premise of rule $(Sto2)$: the reason being that $(q) | (\nu q' \rightarrow \lambda')(q')$ can be rearranged to $(\nu q' \rightarrow \lambda')((q) | (q'))$. Moreover, since we consider stochastic $P \longrightarrow \sigma$ transitions only for terms $P \in \mathcal{P}_{ext}$ such that there exists $P' \in \mathcal{P}$ with $P \equiv P'$ (as we will see, all terms P that are reachable by transitions of the operational semantics from an initial term in \mathcal{P} are in this form), we have that multiple outgoing stochastic transitions for the same P cannot be produced. This is a consequence of the fact that in such terms P each stochastic binder binds at most one (q) prefix (and of the fact that the “partial” stochastic transitions mentioned above cannot be produced). Summing up, we have that for such terms P , either no $P \longrightarrow \sigma$ transition is generated or a single $P \longrightarrow \sigma$ transition is generated. In the latter case it correctly evaluates the multiset of Markovian transitions leaving a state on the basis of symbolic delay transitions, each of which (since all stochastic binders must have been extruded up to the outermost level) is guaranteed to have a different name. All these facts, for which we gave here just an informal intuition, will be proved formally in the following (Propositions 3.1, 3.2, 3.3 and the Harmony Theorem 3.1).

Example 3.1. Consider $P = (5).A | (5).A$ and $A \triangleq (5).A$. We have that, according to the standard labeled semantics presented in Section 2, P performs transitions:

$$\begin{array}{l}
P \xrightarrow{5, |_0^\bullet} A | (5).A \\
P \xrightarrow{5, |_1^\bullet} (5).A | A
\end{array}$$

Let us now apply our reduction semantics to P . We have that

$$\begin{aligned}
P &\equiv (\nu q \rightarrow 5)(q).A | (\nu q \rightarrow 5)(q).A \equiv \\
&\quad (\nu q \rightarrow 5)(\nu q' \rightarrow 5)((q).A | (q').A)
\end{aligned}$$

thus P performs the stochastic transition

$$P \longrightarrow \sigma = \{(\lambda, [(5).A | (5).A]_{\equiv}), (5, [(5).A | (5).A]_{\equiv})\}$$

that is a multiset including two instances of the same pair.

This is because, after applying (Sto1) and (Sto2), we get $P \longrightarrow \sigma'$ with

$$\begin{aligned}\sigma' &= \{ \{ (5, [(\nu q \rightarrow 5)(\nu q' \rightarrow 5)(A|(q').A)]_{\equiv}), (5, [(\nu q \rightarrow 5)(\nu q' \rightarrow 5)((q).A|A)]_{\equiv}) \} \\ &= \{ \{ (5, [(\nu q' \rightarrow 5)(A|(q').A)]_{\equiv}), (5, [(\nu q \rightarrow 5)((q).A|A)]_{\equiv}) \} \\ &= \{ \{ (5, [A|(5).A]_{\equiv}), (5, [(5).A|A]_{\equiv}) \} \} = \sigma\end{aligned}$$

The example shows that, at every (stochastic) transition, stochastic binders are moved:

- first inside-out (from prefixes to the top-level) to make it possible to apply (Sto1) and (Sto2) rules;
- then outside-in (back, so to be eliminated or become prefixes again) in order to obtain again a representative in \mathcal{P} for each of the reached congruence classes.

This shows that, differently from other uses of “unique names” in the literature (as e.g. for expressing ST semantics [6]), where the purpose of the generated names is to uniquely identify elements across multiple subsequent transitions, here names are generated for the sole purpose of uniquely identifying single transitions and can be forgotten after the transition itself is performed.

The transition system of terms can be represented in a *finitary* way by taking states to be congruence classes $[P]_{\equiv} \in \mathcal{P}_{ext}/\equiv$ (recall that, due to standard laws like $P|0 \equiv P$, every term has an infinite number of congruent terms). This will be important when deriving, as done in Section 2 for the labeled operational semantics, the underlying Interactive Markov Chain.

Definition 3.3. Let $P, P' \in \mathcal{P}_{ext}$. We define $[P]_{\equiv} \xrightarrow{\hat{q}} [P']_{\equiv}$ whenever $P \xrightarrow{\hat{q}} P'$. Let $P \in \mathcal{P}_{ext}$ and $\sigma \in RDist(\mathcal{P}_{ext}/\equiv)$. We define $[P]_{\equiv} \longrightarrow \sigma$ whenever $P \longrightarrow \sigma$.

The above is well-defined because congruent terms P have the same outgoing transitions (rule of closure w.r.t. congruence in each of Tables 4 and 5).

Notice that, it would have also been possible to define directly transitions on equivalence classes, instead of resorting to Definition 3.3, by putting equivalence classes inside rules, e.g. for parallel

$$\frac{[P]_{\equiv} \xrightarrow{\hat{q}} [P']_{\equiv}}{[P|Q]_{\equiv} \xrightarrow{\hat{q}} [P'|Q]_{\equiv}}$$

This would have also made the two rules of closure w.r.t. congruence in each of Tables 4 and 5 superfluous (working directly with classes yields the same effect). For the sake of clarity and readability, we preferred however to stick to the more standard approach of defining semantics via rules of closure, which also avoids burdening all other rules with the equivalence class notation around terms.

3.3. Properties and Results

Let us now analyze properties of (outgoing) transitions of terms $P \in \mathcal{P}$, that is, terms belonging to the non-extended syntax of Definition 2.1. Notice that, since congruent terms have the same outgoing transitions, such properties obviously hold also for terms $P \in \mathcal{P}_{ext}$ such that $P \equiv P'$ for some $P' \in \mathcal{P}$.

The following proposition says that terms P of the non-extended syntax can just perform standard reductions and stochastic transitions. This means that, for such terms P , the \xrightarrow{q} transitions are used only during inference of stochastic transitions.

Proposition 3.1. *Let $P \in \mathcal{P}$. There is no $P' \in \mathcal{P}_{ext}$, $q \in \mathcal{QN}$ such that $P \xrightarrow{q} P'$.*

Proof. A direct consequence of the fact that for any $Q \equiv P$, with $P \in \mathcal{P}$, we have that, due to the laws of \equiv , all delay prefixes (q) occurring in Q are bound (rules for binders do not allow transitions labeled with the bound name to be inferred). \square

We now present a crucial property, which we informally discussed in Section 3.2, of our reduction semantics that shows that the semantics of stochastic transitions is well-defined: if a $P \in \mathcal{P}$ term has an outgoing stochastic transition $P \longrightarrow \sigma$ then it is unique. Notice that, besides the intuitive observations we already made in Section 3.2, this also entails proving that, no matter the order in which we choose to solve stochastic binders, we get a stochastic transition with the same σ (this is one of the sources of the complexity of the proof).

Proposition 3.2. *Let $P \in \mathcal{P}$ and $P \longrightarrow \sigma$. We have $P \longrightarrow \sigma'$ implies $\sigma = \sigma'$.*

Proof. The proof, reported in Appendix A, essentially shows that, supposing n to be the number of Markovian delays “immediately performable” by P and q_1, \dots, q_n any set of distinguished names, there exists P' such that $P \equiv (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) P'$ and $\forall h, 1 \leq h \leq n. P' \xrightarrow{q_h} P'_h$. Moreover for any such P' we have:

$$\sigma = \sigma' = \{ \{ (\lambda_1, [(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) P'_1]_{\equiv}) \} \cup \dots \cup \{ (\lambda_n, [(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) P'_n]_{\equiv}) \} \}$$

\square

Finally we show that for any $P' \in \mathcal{P}_{ext}$ that is reachable by performing transitions from a term $P \in \mathcal{P}$ (i.e. a term belonging to the non-extended syntax) it is always possible to find a term $P'' \in \mathcal{P}$ that is congruent to P' .

Proposition 3.3. *Let $P \in \mathcal{P}$ and $P' \in \mathcal{P}_{ext}$ such that: $P \longrightarrow P'$ or $P \longrightarrow \sigma \wedge \exists \lambda: (\lambda, [P']_{\equiv}) \in \sigma$. There exists $P'' \in \mathcal{P}$ such that $P'' \equiv P'$.*

Proof. For any $Q \equiv P$ with $P \in \mathcal{P}$ we have: all delay prefixes (q) occurring in Q are bound and every stochastic binder can bind at most a single delay prefix (q). It is easy to verify that this property is preserved by reductions $Q \longrightarrow Q'$ or transitions $Q \longrightarrow \sigma$ for which $\exists \lambda$ such that $(\lambda, [Q']_{\equiv}) \in \sigma$, i.e. for any such Q' the same property holds true. For instance performing the σ transition causes the corresponding stochastic binder to not bind any delay prefix in the target state. Any Q' satisfying the property above can be shown to be congruent to a term of P by just removing stochastic binders not binding any delay prefix and by turning stochastic binder binding a single delay prefix (q) into a Markovian delay prefix. \square

A consequence of the above propositions and of Definition 3.3 is that, given $P \in \mathcal{P}$, we can represent the semantics of P as a transition system whose *initial state* is $[P]_{\equiv}$ and, in general, *states* (congruence classes reachable from $[P]_{\equiv}$) are $[P']_{\equiv}$ such that $P' \in \mathcal{P}$: such classes always have a representative in \mathcal{P} .

Concerning the underlying Interactive Markov Chain, we have:

- the rate $\gamma([P]_{\equiv}, [P']_{\equiv})$ of transiting from state $[P]_{\equiv}$ to state $[P']_{\equiv}$ is:
 $\sum_{(\lambda, [P']_{\equiv}) \in \sigma} \lambda \cdot \mu_{\sigma}(\lambda, [P']_{\equiv})$ if $[P]_{\equiv} \longrightarrow \sigma$ for some σ ; 0 otherwise³
- τ transitions of the Interactive Markov Chain are standard reductions.

The above also provides a definition of bisimulation over states of $P \in \mathcal{P}$ by applying it to Definition 2.2 (and to other equivalences in [16]). Obviously, w.r.t. the Interactive Markov Chain obtained by labeled operational semantics, reduction semantics of a term P yields a *closed system interpretation*, i.e. as if all (free) actions of P were bound at the top level. Therefore the obtained Interactive Markov Chain is just labeled with τ actions as for all Interactive Markov Chains corresponding to complete systems, which are commonly the ones to be analyzed for performance.

We now formalize this correspondence. This has been done for the π -calculus, see, e.g., Harmony Lemma in [22] or [19]. Here we extend this result to Markovian delays, thus also showing correctness of the stochastic binder idea: in particular that the semantics of stochastic transitions correctly deals with rate multiplicity of identical transitions and that all delays are actually accounted for when calculating the stochastic transition of a term (see discussion about “partial” stochastic transitions in Section 3.2).

Theorem 3.1 (Harmony). *Let $P \in \mathcal{P}$. We have:*

- $P \longrightarrow P' \text{ iff } P \xrightarrow{\tau} \equiv P'$
- $P \longrightarrow \sigma \text{ for some } \sigma \text{ iff } P \xrightarrow{\lambda, id} P' \text{ for some } \lambda, id \text{ and } P'. \text{ Moreover, when}$

³We assume an empty sum to yield 0. Given a multiset m , $\mu_m(e)$ denotes the number of occurrences of element e in m .

both hold, we have that $\forall P' \in \mathcal{P}$

$$\sum_{(\lambda, [P']_{\equiv}) \in \sigma} \lambda \cdot \mu_{\sigma}(\lambda, [P']_{\equiv}) = \sum_{P \xrightarrow{\lambda, id} \equiv P'} \lambda$$

Proof. The proof, reported in Appendix A is based on that of Proposition 3.2: in particular on the structure, we previously presented, of the unique σ (if any) such that $P \rightarrow \sigma$ and on the following observations. Assume q_1, \dots, q_n to be any set of distinguished names. Given Q including unique occurrences of prefixes $(q_1), \dots, (q_n)$ in its “immediate behaviour” and such that $P \equiv Q\{\lambda_i/q_i \mid 1 \leq i \leq n\}$, we have $P \equiv (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n)Q$ and $\forall h, 1 \leq h \leq n. Q \xrightarrow{q_h} Q'_h$. Moreover, for each index h , with $1 \leq i \leq h$, and Q''_h such that $Q\{\lambda_i/q_i \mid 1 \leq i \leq n\} \xrightarrow{\lambda_h, id_h} Q''_h$ there exists P'_h , with $P'_h \equiv Q''_h$, such that $P \xrightarrow{\lambda_h, id_h} P'_h$; and vice versa. The proof shows that both sides of the theorem equation, thus, yield $\sum_{h \in I} \lambda_h$, with I being the set of indexes h such that $Q''_h \equiv P'$. \square

4. Markovian Actions

We now consider the two variants, we discussed in the introduction, extending π -calculus with rates attached to actions: for each of them we present a labeled operational semantics using the classical approach of [21] and extensions of our technique based on stochastic binders and stochastic names to produce a reduction semantics. With respect to Markovian delays, having rates attached to actions complicates significantly the problem of developing a (correct) semantics in that Markovian transitions are no longer simply generated from standalone prefixes, but are produced by synchronization of an output and an input (under a specific rule for calculating the rate of the τ transition obtained by synchronization). In both variants that we consider output and τ prefixes are endowed with a rate: their difference stems from the way inputs are dealt with (and in how the synchronization rule mentioned above is defined). In the simpler variant input prefixes are not explicitly quantified, while in the other one they are endowed with a weight. Similarly to what we did for Markovian delays, for both versions we will: study the properties of the presented reduction semantics, show how to derive the underlying Markov Chain, and compare them with the standard labeled semantics.

4.1. Syntax

In π -calculus with Markovian Actions we extend the syntax of output $\bar{x}\langle y \rangle$ and τ prefixes by attaching a Markovian rate λ to them: for the sake of simplicity we just represent it as a subscript, i.e. $\bar{x}\langle y \rangle_{\lambda}$ and τ_{λ} . Concerning input prefixes, we consider two versions:

- *non-quantified inputs*, in which input prefixes are not explicitly quantified (their syntax is unmodified w.r.t. π -calculus). The idea is that the τ transition resulting from the synchronization of a λ -decorated output on channel “ a ” and an input on “ a ” simply has rate λ : this corresponds to multiplying λ by the number of possible synchronizations over “ a ”, similarly to what is done in the context of biochemistry, see e.g. [8].
- *weighted inputs*, in which input prefixes are endowed with weights $w \in \mathbb{R}^+$, i.e. numbers establishing the probability to choose an input on channel “ a ” given that “ a ” is the channel of the selected output, as for the Stochastic π -calculus [21] with inputs having (weight equipped) unspecified rates [17, 21].

We take the two above versions of π -calculus with Markovian actions to have the same syntax as π -calculus with Markovian delays (Definition 2.1), apart from redefinition of prefixes α . We denote: with \mathcal{P}^{ra} (pointing out that now we have rates λ attached to actions, instead of having them in separate prefixes) the set of terms of π -calculus with Markovian actions and non-quantified inputs; with \mathcal{P}^{wra} (pointing out that we additionally have weights) the set of terms of π -calculus with Markovian actions and weighted inputs.

Definition 4.1. *We take \mathcal{P}^{ra} to be the set of terms P, Q, \dots generated by*

$$\begin{aligned} P &::= M \mid P \mid P' \mid (\nu x)P \mid A(x_1, \dots, x_n) \\ M &::= \underline{0} \mid \alpha.P \mid M + M' \\ \alpha &::= \bar{x} \langle y \rangle_\lambda \mid x(y) \mid \tau_\lambda \end{aligned}$$

We take \mathcal{P}^{wra} to be the set of terms P, Q, \dots generated by the same syntax, where in the definition of prefixes α we consider inputs $x(y)_w$, instead of $x(y)$.

In the following we use $\gamma \in \mathbb{R}^+$ to range over both rates λ and weights w .

4.2. Classical Approach

We now define the labeled operational semantics of the two considered versions of π -calculus with Markovian Actions, using the classical approach based on (operator dependent) identifiers.

4.2.1. Non-quantified Inputs

With respect to Section 2, we now have a single (combined) kind of transition, that is Markovian and labeled transitions $P \xrightarrow{v, id} P'$, where v is any of $\bar{x}y_\lambda$, $\bar{x}(y)_\lambda$, xy or τ_λ and id is the identifier used to distinguish identical transitions (in order to correctly account for their multiplicity). The identification based approach of Section 2 must, thus, be now extended to also represent Markovian transitions obtained by synchronization. We do this by enriching the syntax of identifiers id with the construct $\langle \mid_0 id_1, \mid_1 id_2 \rangle$, where id_1, id_2 are themselves identifiers, see [21]. Thus the syntax of identifiers is now:

$$id ::= \bullet \mid +_0 id \mid +_1 id \mid \mid_0 id \mid \mid_1 id \mid \langle \mid_0 id, \mid_1 id \rangle$$

$$\begin{array}{c}
\tau_\lambda.P \xrightarrow{\tau_\lambda, \bullet} P \quad \bar{x}\langle y \rangle_\lambda.P \xrightarrow{\bar{x}y_\lambda, \bullet} P \quad x(y).P \xrightarrow{xy', \bullet} P\{y'/y\} \\
\\
\frac{P \xrightarrow{v, id} P'}{P + Q \xrightarrow{v, +_0 id} P'} \quad \frac{P \xrightarrow{v, id} P'}{P|Q \xrightarrow{v, |_0 id} P'|Q} \quad \text{bn}(\mu) \cap \text{fn}(Q) = \emptyset \\
\\
\frac{P \xrightarrow{\bar{x}y_\lambda, id_1} P' \quad Q \xrightarrow{xy, id_2} Q'}{P|Q \xrightarrow{\tau_\lambda, \langle |_0 id_1, |_1 id_2 \rangle} P'|Q'} \quad \frac{P \xrightarrow{\bar{x}(y)_\lambda, id_1} P' \quad Q \xrightarrow{xy, id_2} Q'}{P|Q \xrightarrow{\tau_\lambda, \langle |_0 id_1, |_1 id_2 \rangle} (\nu y)(P'|Q')} \quad y \notin \text{fn}(Q) \\
\\
\frac{P \xrightarrow{\eta} P'}{(\nu x)P \xrightarrow{\eta} (\nu x)P'} \quad x \notin n(\eta) \quad \frac{P \xrightarrow{\bar{x}y_\lambda, id} P'}{(\nu y)P \xrightarrow{\bar{x}(y)_\lambda, id} P'} \quad y \neq x \\
\\
\frac{P\{\tilde{y}/\tilde{x}\} \xrightarrow{\eta} P'}{A(\tilde{y}) \xrightarrow{\eta} P'} \quad A(\tilde{x}) \triangleq P
\end{array}$$

Table 6: Standard Rules for Markovian Actions (Symmetric Rules Omitted)

We present in Table 6 the labeled operational semantics of π -calculus with Markovian actions and non-quantified inputs (over \mathcal{P}^{ra} terms of Definition 4.1). We take η to range over transition labels, i.e. pairs “ v, id ”. Essentially, w.r.t. the semantics in Tables 1 and 2 for the version with Markovian delays, here we apply λ quantification to all output and τ prefixes/labels and we associate an identifier id to all transitions: as in [21] we use $\langle |_0 id_1, |_1 id_2 \rangle$ as identifier for the transition inferred by synchronization where id_1 and id_2 are the identifiers of the synchronizing transitions (for choice and unsynchronized parallel we use the same identifier constructions as those in Table 2).

Example 4.1. Consider $P = \bar{x}\langle y \rangle_2.P'|(x(z).P''|x(z').P''')$. We have that P performs transitions:

$$\begin{aligned}
P &\xrightarrow{\tau_2, \langle |_0 \bullet, |_1 |_0 \bullet \rangle} P_1 = P'|(P''\{y/z\}|x(z').P''') \\
P &\xrightarrow{\tau_2, \langle |_0 \bullet, |_1 |_1 \bullet \rangle} P_2 = P'|(x(z).P''|P'''\{y/z'\})
\end{aligned}$$

4.2.2. Weighted Inputs

We present in Table 7 the labeled operational semantics of π -calculus with Markovian actions and weighted inputs (over \mathcal{P}^{wra} terms of Definition 4.1). Essentially, w.r.t. that in Table 6 for the version with non-quantified inputs, here we attach a weight w to all input prefixes/labels and we replace the rate λ of the τ_λ action inferred in the two rules for action synchronization with rate $(\lambda \cdot w)/W_Q^{xy}$, where W_Q^{xy} is the sum of the weights of all input transitions xy (i.e. those synchronizable with the output transition $\bar{x}y$ considered) that Q can perform.

$$\begin{array}{c}
\tau_{\lambda}.P \xrightarrow{\tau_{\lambda}, \bullet} P \quad \bar{x} \langle y \rangle_{\lambda}.P \xrightarrow{\bar{x}y_{\lambda}, \bullet} P \quad x(y)_w.P \xrightarrow{xy'_w, \bullet} P\{y'/y\} \\
\\
\frac{P \xrightarrow{v, id} P'}{P + Q \xrightarrow{v, +_0 id} P'} \quad \frac{P \xrightarrow{v, id} P'}{P|Q \xrightarrow{v, |_0 id} P'|Q} \quad \text{bn}(\mu) \cap \text{fn}(Q) = \emptyset \\
\\
\frac{P \xrightarrow{\bar{x}y_{\lambda}, id_1} P' \quad Q \xrightarrow{xy_w, id_2} Q'}{P|Q \xrightarrow{\tau_{(\lambda \cdot w)/W_Q^{xy}, \langle |_0 id_1, |_1 id_2 \rangle}} P'|Q'} \quad \frac{P \xrightarrow{\bar{x}(y)_{\lambda}, id_1} P' \quad Q \xrightarrow{xy_w, id_2} Q'}{P|Q \xrightarrow{\tau_{(\lambda \cdot w)/W_Q^{xy}, \langle |_0 id_1, |_1 id_2 \rangle}} (\nu y)(P'|Q')} \quad y \notin \text{fn}(Q) \\
\\
\frac{P \xrightarrow{\eta} P'}{(\nu x)P \xrightarrow{\eta} (\nu x)P'} \quad x \notin n(\eta) \quad \frac{P \xrightarrow{\bar{x}y_{\lambda}, id} P'}{(\nu y)P \xrightarrow{\bar{x}(y)_{\lambda}, id} P'} \quad y \neq x \\
\\
\frac{P\{\tilde{y}/\tilde{x}\} \xrightarrow{\eta} P'}{A(\tilde{y}) \xrightarrow{\eta} P'} \quad A(\tilde{x}) \triangleq P
\end{array}$$

where: $W_Q^{xy} = \sum_Q \xrightarrow{xy_{w'}, id} w'$

Table 7: Standard Rules for Markovian Actions with Weighted Inputs (Symm. Rules Omitted)

As we already mentioned, the obtained semantics is like that of the Stochastic π -calculus [21] with weights being dealt with as (weight equipped) unspecified rates of [17, 21] and W_Q^{xy} in Table 7 being, in the terminology of [17, 21], the *apparent rate* of xy in Q .

Example 4.2. Consider $P = \bar{x} \langle y \rangle_3.P'|(x(z)_4.P''|(x(z')_8.P'''))$, i.e. the example we already mentioned in the introduction. We have that P performs transitions:

$$\begin{aligned}
P &\xrightarrow{\tau_1, \langle |_0 \bullet, |_1 |_0 \bullet \rangle} P_1 = P'|(P''\{y/z\}|x(z')_8.P''') \\
P &\xrightarrow{\tau_2, \langle |_0 \bullet, |_1 |_1 \bullet \rangle} P_2 = P'|(x(z)_4.P''|P'''\{y/z'\})
\end{aligned}$$

4.2.3. Bisimilarity

Similarly to what we did for Markovian delays in Section 2, we now discuss how bisimulation is defined for π -calculus with Markovian actions, encompassing both the case of weighted and non-quantified inputs.

We first need to modify the $\gamma(P, P')$ function considered in Section 2, yielding the total rate of transiting from state P to P' , to take into account that now Markovian transitions are labeled. Given a standard π -calculus label μ (see Section 2), we take, as in [17, 1], $\gamma(P, \mu, P')$ to be defined as:

$$\gamma(P, \mu, P') = \sum_{P \xrightarrow{\mu \gamma, id} P'} \gamma$$

where, for the version without weights, we assume μ_γ to be xy if μ is xy and $\gamma = 1$. That is, in the case μ is an output label, $\gamma(P, \mu, P')$ yields the total rate of transiting from state P to P' via μ labeled transitions. In the case μ is, instead, an input label: for the version without weights $\gamma(P, \mu, P')$ yields the number of μ labeled transitions going from P to P' ; for the version with weights it yields the total weight of transiting from state P to P' via μ labeled transitions.

Let also, as usual, $\gamma(P, \mu, Set) = \sum_{P' \in Set} \gamma(P, \mu, P')$.

We now consider Markovian bisimulation as in [17, 1]. In the following definition we use $\mathcal{P}^{[w]ra}$ to mean that it holds both when we take it to be \mathcal{P}^{ra} and, also, when we instead consider \mathcal{P}^{wra} .

Definition 4.2. *An equivalence relation \mathcal{B} on $\mathcal{P}^{[w]ra}$ is a strong bisimulation iff $P \mathcal{B} Q$ implies for all μ and all equivalence classes C of \mathcal{B} :*

$$\gamma(P, \mu, C) = \gamma(Q, \mu, C)$$

Two processes P and Q are strongly bisimilar if (P, Q) is contained in some strong bisimulation \mathcal{B} .

4.3. Reduction Semantics

We now define the semantics of the two considered versions of π -calculus with Markovian actions in *reduction style*.

4.3.1. Non-quantified Inputs

We first extend the syntax of terms $P, Q, \dots \in \mathcal{P}^{ra}$ (Definition 4.1) by *adding* stochastic names and stochastic binders. We have that a stochastic name q can be associated with a π -calculus output or input prefix. Moreover, we have output stochastic binders, which carry the quantification λ for a name q of an output prefix, and input stochastic binders, which are, instead, non-quantified. Moreover τ prefixes are given a pair of stochastic names q, q' as if they were obtained (as for τ transitions) by the synchronization between an output and an input, with the output stochastic binder carrying the quantification λ of the τ (and the input one being non-quantified).

Definition 4.3. *We take \mathcal{P}_{ext}^{ra} to be the set of terms P, Q, \dots generated by extending the syntax of \mathcal{P}^{ra} terms in Definition 4.1 as follows:*

$$\begin{aligned} P &::= \dots \mid (\nu q \rightarrow \lambda)P \mid (\nu q)P \\ \alpha &::= \dots \mid \bar{x} \langle y \rangle_q \mid x(y)_q \mid \tau_{q, q'} \end{aligned}$$

Concerning structural congruence, in addition to standard laws in Table 3, we now consider:

$$\begin{aligned} \bar{x} \langle y \rangle_\lambda.P + M &\equiv (\nu q \rightarrow \lambda)(\bar{x} \langle y \rangle_q.P + M) && \text{if } q \notin \text{fn}(M, P) \\ x(y).P + M &\equiv (\nu q)(x(y)_q.P + M) && \text{if } q \notin \text{fn}(M, P) \\ \tau_\lambda.P + M &\equiv (\nu q')(\nu q \rightarrow \lambda)(\tau_{q, q'}.P + M) && \text{if } q, q' \notin \text{fn}(M, P) \end{aligned}$$

The idea behind the semantics is an elaboration of that we presented for Markovian delays. As before, in order for $P \longrightarrow \sigma$ to be generated, term P must be rearranged by structural congruence in such a way that all its stochastic binders (those actually binding a prefix) are the outermost operators, i.e. they must be all extruded up to the top level. However now, as we will see, we also need to force binders of inputs to be put outside binders of outputs they can synchronize with. Only under this condition $P \longrightarrow \sigma$ are *generated from the output binders*.

The reduction semantics is defined in terms of *two* kind of transitions: symbolically quantified reductions $P \xrightarrow{\hat{q}} P'$, with $\hat{q} \in \mathcal{QN} \cup \{\varepsilon\}$, defined in Table 8 (which take the place of standard reductions and symbolically quantified transitions of case of Markovian delays in Section 3.2); and stochastic reductions $P \longrightarrow \sigma$, with $\sigma \in RDist(\mathcal{P}_{ext}^a / \equiv)$, defined in Table 9.⁴ $P \xrightarrow{q, \hat{q}} P'$ represent reductions (either execution of τ prefixes or synchronizations) to which symbolic quantification is attached by specifying the stochastic name of an output followed by that of an input (or ε). In the case $P \xrightarrow{q, \hat{q}'} P'$ traverses the input binder $(\nu q')$ then we get an ε as input quantification label, i.e. $(\nu q')P \xrightarrow{q, \varepsilon} (\nu q')P'$. This is obtained by assuming, in Table 8, $\hat{q} \setminus \theta$ to be: \hat{q} if $\theta \neq \hat{q}$; ε if $\theta = \hat{q}$. $P \xrightarrow{q, \hat{q}'} P'$ is instead not allowed to traverse an output binder $(\nu q \rightarrow \lambda)$ for the output stochastic name q .

The latter input binder traversal mechanism is needed to detect the case in which the input stochastic binder for q' is put (by congruence relation) inside the output stochastic binder for q : in this case the input quantification label becomes ε . This is exploited in the semantics of stochastic reductions (Table 9) to prevent production of the stochastic reduction by the output stochastic binder, because, in order to correctly carry out its counting work, it needs the stochastic names of all the inputs it can synchronize with (thus this forces all their stochastic binders to not be put inside the output stochastic binder).

All these facts, for which we gave here just an informal intuition, will be proved formally in the following (Propositions 4.1, 4.2, 4.3 and the Harmony Theorem 4.1).

Example 4.3. Consider again term $P = \bar{x} < y >_2 . P' | (x(z) . P'' | x(z') . P''')$ of Example 4.1. Let us now apply our reduction semantics to P . We have that

$$P \equiv (\nu q')(\nu q'')(\nu q \rightarrow 2)(\bar{x} < y >_q . P' | (x(z)_{q'} . P'' | x(z')_{q''} . P'''))$$

thus P performs the stochastic transition

$$P \longrightarrow \{ (2, [P_1]_{\equiv}), (2, [P_2]_{\equiv}) \}$$

where P_1 and P_2 are those of Example 4.1.

⁴We assume, similarly to $(\nu q \rightarrow \lambda)\sigma$, that $(\nu q)\sigma = \{ (\lambda, [(\nu q)P]_{\equiv}) \mid (\lambda, [P]_{\equiv}) \in \sigma \}$.

$$\begin{array}{c}
\tau_{q,q'}.P + M \xrightarrow{q,q'} P \quad \bar{x} < y >_q . P + M \mid x(z)_{q'}.Q + N \xrightarrow{q,q'} P \mid Q\{y/z\} \\
\\
\frac{P \xrightarrow{q,\hat{q}} P'}{P \mid Q \xrightarrow{q,\hat{q}} P' \mid Q} \quad \frac{P \xrightarrow{q,\hat{q}} P'}{(\nu\theta \rightarrow \hat{\lambda})P \xrightarrow{q,\hat{q} \setminus \theta} (\nu\theta \rightarrow \hat{\lambda})P'} \quad \theta \neq q \\
\\
\frac{P \equiv Q \quad Q \xrightarrow{q,\hat{q}} Q' \quad Q' \equiv P'}{P \xrightarrow{q,\hat{q}} P'}
\end{array}$$

Table 8: Standard Reduction Rules with Symbolic Quantification Added

$$\begin{array}{c}
(StoOut1) \frac{P \longrightarrow \sigma \quad \varepsilon \notin \{\hat{q} \mid P \xrightarrow{q,\hat{q}}\} \neq \emptyset}{(\nu q \rightarrow \lambda)P \longrightarrow (\nu q \rightarrow \lambda)\sigma \cup \{(\lambda, [(\nu q \rightarrow \lambda)P_{q'}]_{\equiv}) \mid P \xrightarrow{q,q'} P_{q'}\}} \\
(StoOut2) \frac{\nexists \sigma : P \longrightarrow \sigma \quad \varepsilon \notin \{\hat{q} \mid P \xrightarrow{q,\hat{q}}\} \neq \emptyset}{(\nu q \rightarrow \lambda)P \longrightarrow \{(\lambda, [(\nu q \rightarrow \lambda)P_{q'}]_{\equiv}) \mid P \xrightarrow{q,q'} P_{q'}\}} \\
(StoIn) \frac{P \longrightarrow \sigma}{(\nu q)P \longrightarrow (\nu q)\sigma} \quad (StoCong) \frac{P \equiv Q \quad Q \longrightarrow \sigma}{P \longrightarrow \sigma}
\end{array}$$

Table 9: Stochastic Reduction Rules for Markovian Actions

4.3.2. Weighted Inputs

We consider an extension of the syntax of $P, Q, \dots \in \mathcal{P}^{wra}$ terms (Definition 4.1) similar to the one we did, in Definition 4.3, for $P, Q, \dots \in \mathcal{P}^{ra}$ terms. The only difference is that now stochastic binders for inputs include quantification with a weight w , i.e. they are denoted by $(\nu q \rightarrow w)$.

Definition 4.4. We take \mathcal{P}_{ext}^{wra} to be the set of terms P, Q, \dots generated by extending the syntax of \mathcal{P}^{wra} terms in Definition 4.1 as follows:

$$\begin{array}{lcl}
P & ::= & \dots \mid (\nu q \rightarrow \lambda)P \mid (\nu q \rightarrow w)P \\
\alpha & ::= & \dots \mid \bar{x} < y >_q \mid x(y)_q \mid \tau_{q,q'}
\end{array}$$

Concerning structural congruence, we consider, in addition to standard ones in Table 3, the same three laws we had for \mathcal{P}_{ext}^{ra} terms with the only difference that now input prefixes and stochastic input binders are endowed with a weight w . The additional laws are thus as follows.

$$\begin{array}{lll}
\bar{x} < y >_\lambda . P + M & \equiv & (\nu q \rightarrow \lambda)(\bar{x} < y >_q . P + M) \quad \text{if } q \notin \text{fn}(M, P) \\
x(y)_w . P + M & \equiv & (\nu q \rightarrow w)(x(y)_q . P + M) \quad \text{if } q \notin \text{fn}(M, P) \\
\tau_\lambda . P + M & \equiv & (\nu q' \rightarrow w)(\nu q \rightarrow \lambda)(\tau_{q,q'} . P + M) \quad \text{if } q, q' \notin \text{fn}(M, P)
\end{array}$$

Notice that in the last law, where, from a τ prefix, we generate stochastic names and the respective stochastic binders for a pair of (fictitious) output and input,

the choice of the weight w is not significant, in that q' is the only input that can synchronize with the q output.

The idea behind the semantics is an elaboration of that we presented for the case of Markovian actions with non-quantified inputs. As before, in order for $P \longrightarrow \sigma$ to be generated, term P must be rearranged by structural congruence in such a way that all its stochastic binders (those actually binding a prefix) are the outermost operators, with binders of inputs being outside binders of outputs they can synchronize with: under this condition, stochastic reductions $P \longrightarrow \sigma$ are generated from the output binders. However now, since the input binders carry the weight information, rates included in the generated σ are represented symbolically by expressions over input stochastic names (standing for their weight): we call such a σ a *symbolic rate distribution*. The generated σ will be, then, turned into a usual rate distribution (i.e. a numerical one, of the kind we considered before) by traversing all the input binders ($\nu q \rightarrow w$): each of them replaces the name q with the weight w inside expressions and (partially) evaluates them.

We, thus, start by introducing symbolic rate distributions to be multisets of pairs where the first element, instead of being just a rate $\lambda \in \mathbf{R}^+$ (as in rate distributions of Definition 3.2), is an expression E defined as follows.

Definition 4.5. *We take \mathcal{E} to be the set of expressions E generated by*

$$\begin{aligned} E &::= \sum_{i \in I} V_i \mid E \cdot E \mid E/E \\ V &::= q \mid \gamma \end{aligned}$$

when writing expressions we assume $\sum_{i \in I} V_i$ with $I = \{k\}$ to be simply denoted by V_k .

We now define symbolic rate distributions. Since a rate distribution (Definition 3.2) is a particular case of a symbolic rate distribution (when all expressions E it includes are such that $E = \gamma$ for some γ) we just extend the use of the metavariable σ to also range over symbolic rate distributions.

Definition 4.6. *A symbolic rate distribution σ over a countable set S is a non-empty finite multiset over $\mathcal{E} \times S$, i.e. a multiset of pairs (E, e) , with $E \in \mathcal{E}$ and $e \in S$. We use $SRDist(S)$ to denote the set of symbolic rate distributions over S .*

Reduction semantics is defined, as in the case of Markovian actions with non-quantified inputs, in terms of two kind of transitions: symbolically quantified reductions $P \xrightarrow{\hat{q}, \hat{q}} P'$, with $\hat{q} \in \mathcal{QN} \cup \{\varepsilon\}$, that are unchanged, i.e. still defined as in Table 8; and stochastic reductions $P \longrightarrow \sigma$, with $\sigma \in SRDist(\mathcal{P}_{ext}^{wra} / \equiv)$, defined in the new Table 10.⁵ With respect to Table 9 considered before for

⁵We assume, similarly to Table 9 that $(\nu q \rightarrow \lambda)\sigma = \{(E, [(\nu q \rightarrow \lambda)P]_{\equiv}) \mid (E, [P]_{\equiv}) \in \sigma\}$ and $(\nu q \rightarrow w)\sigma = \{(E, [(\nu q \rightarrow w)P]_{\equiv}) \mid (E, [P]_{\equiv}) \in \sigma\}$.

$$\begin{array}{c}
(StoOut1) \frac{P \longrightarrow \sigma \quad \varepsilon \notin \mathcal{Q} = \{\widehat{q} \mid P \xrightarrow{q, \widehat{q}}\} \neq \emptyset}{(\nu q \rightarrow \lambda)P \longrightarrow (\nu q \rightarrow \lambda)\sigma \cup \{((\lambda \cdot q')/\sum \mathcal{Q}, [(\nu q \rightarrow \lambda)P_{q'}]_{\equiv}) \mid P \xrightarrow{q, q'} P_{q'}\}} \\
(StoOut2) \frac{\exists \sigma: P \longrightarrow \sigma \quad \varepsilon \notin \mathcal{Q} = \{\widehat{q} \mid P \xrightarrow{q, \widehat{q}}\} \neq \emptyset}{(\nu q \rightarrow \lambda)P \longrightarrow \{((\lambda \cdot q')/\sum \mathcal{Q}, [(\nu q \rightarrow \lambda)P_{q'}]_{\equiv}) \mid P \xrightarrow{q, q'} P_{q'}\}} \\
(StoIn) \frac{P \longrightarrow \sigma}{(\nu q \rightarrow w)P \longrightarrow (\nu q \rightarrow w)(\sigma[w/q])} \quad (StoCong) \frac{P \equiv Q \quad Q \longrightarrow \sigma}{P \longrightarrow \sigma}
\end{array}$$

Table 10: Stochastic Reduction Rules for Markovian Actions with Weighted Inputs

the version with non-quantified inputs, here we replace the rate λ in the rate distribution produced by *(StoOut1)* and *(StoOut2)* rules with the expression $(\lambda \cdot q')/\sum \mathcal{Q}$: the rate λ of the output multiplied by the weight of the specific synchronizing input (represented symbolically by its name q') and divided by the sum of the weights of all the inputs it can synchronize with (represented symbolically by $\sum \mathcal{Q}$). Concerning the latter, we assume $\sum \mathcal{Q}$ to stand for $\sum_{i \in I} q_i$ (belonging to the syntax of expressions E) by considering any name indexing q_i such that $\mathcal{Q} = \{q_1, \dots, q_n\}$. Moreover the rule *(StoIn)* is now changed by considering, in its target, $\sigma[w/q]$ instead of the same distribution σ considered in its premise. In Table 10 we assume $\sigma[w/q]$ to be defined as the symbolic rate distribution obtained by replacing every expression E occurring inside σ with $E[w/q]$, i.e. $\sigma[w/q] = \{(E[w/q], e) \mid (E, e) \in \sigma\}$. In turn we assume $E[w/q]$ to be the expression obtained from E by first replacing occurrences of name q in E with value w and by subsequently computing all its operations that include no names in their arguments. Formally, we define $E[\gamma/q]$ as $eval(E\{\gamma/q\})$ where: $E\{\gamma/q\}$ is the expression E' obtained by syntactically replacing all occurrences of $q \in \mathcal{QN}$ with $\gamma \in \mathbb{R}^+$ inside E , and the *eval* function is defined as follows.

Definition 4.7. *Function eval from \mathcal{E} to \mathcal{E} , that (partially) evaluates an expression E yielding an expression E' , is defined by*

- $eval(\gamma) = \gamma$
- $eval(q) = q$
- $eval(\gamma' \cdot \gamma'') = \gamma$ where γ is the result of $\gamma' \cdot \gamma''$
 $eval(E_1 \cdot E_2) = eval(E_1) \cdot eval(E_2)$ if $\exists \gamma', \gamma'': E_1 = \gamma' \wedge E_2 = \gamma''$
- $eval(\gamma'/\gamma'') = \gamma$ where γ is the result of γ'/γ''
 $eval(E_1/E_2) = eval(E_1)/eval(E_2)$ if $\exists \gamma', \gamma'': E_1 = \gamma' \wedge E_2 = \gamma''$
- $eval(\sum_{i \in I} \gamma_i) = \gamma$ where γ is the result of $\sum_{i \in I} \gamma_i$
 $eval(\sum_{i \in I} V_i) = \sum_{i \in I} V_i$ if $\exists \{\gamma_i \mid i \in I\}: \forall i \in I. V_i = \gamma_i$.

Notice that, since in transition systems we consider stochastic reductions $P \longrightarrow \sigma$ only for terms $P \in \mathcal{P}_{ext}^{wra}$ such that there exists $P' \in \mathcal{P}^{wra}$ with

$P \equiv P'$, we have that $P \longrightarrow \sigma$ implies that σ is a (numerical) rate distribution (Definition 3.2). This is a consequence of the fact that, in such terms P , partial evaluations made by input stochastic binders lead us to completely evaluate the expressions inside the generated symbolic rate distribution once the top-level of term P is reached. This will be proved formally in the following (Proposition 4.2).

Example 4.4. Consider again term $P = \bar{x} < y >_3 . P' | (x(z)_4 . P'' | x(z')_8 . P''')$ of Example 4.2. Let us now apply our reduction semantics to P . We have that

$$P \equiv (\nu q' \rightarrow 4)(\nu q'' \rightarrow 8)(\nu q \rightarrow 3)(\bar{x} < y >_q . P' | (x(z)_{q'} . P'' | x(z')_{q''} . P'''))$$

thus P performs the stochastic transition

$$P \longrightarrow \{(1, [P_1]_{\equiv}), (2, [P_2]_{\equiv})\}$$

where P_1 and P_2 are those of Example 4.2.

4.3.3. Basic Properties

We now discuss basic properties of reduction semantics for π -calculus with Markovian actions presented in previous Sections 4.3.1 and 4.3.2, i.e. encompassing both considered versions: with and without weights. As we did in Section 4.2.3, we use $\mathcal{P}^{[w]ra}$ inside definitions/propositions to mean that the definition/proposition holds both when we take it to be \mathcal{P}^{ra} and, also, when we instead consider \mathcal{P}^{wra} .

We first observe that, as for the case of Markovian delays, the transition system of terms given by the reduction semantics can be represented in a *finitary* way by taking states to be congruence classes $[P]_{\equiv} \in \mathcal{P}_{ext}^{[w]ra} / \equiv$. This is obtained by using the following definition (also here it would have been possible to modify reduction semantics rules so to make them work directly on congruence classes, as explained in Section 3.2).

Definition 4.8. Let $P, P' \in \mathcal{P}_{ext}^{[w]ra}$. We define $[P]_{\equiv} \xrightarrow{q, \hat{q}} [P']_{\equiv}$ whenever $P \xrightarrow{q, \hat{q}} P'$. Let $P \in \mathcal{P}_{ext}^{[w]ra}$ and $\sigma \in SRDist(\mathcal{P}_{ext}^{[w]ra} / \equiv)$. We define $[P]_{\equiv} \longrightarrow \sigma$ whenever $P \longrightarrow \sigma$.

Notice that, in the case of $P \in \mathcal{P}_{ext}^{ra}$, we also have that $\sigma \in RDist(\mathcal{P}_{ext}^{ra} / \equiv)$.

We now analyze properties of (outgoing) transitions of terms $P^{[w]ra} \in \mathcal{P}$, that is, terms belonging to the non-extended syntax, similarly to what we did for the case of Markovian delays in Section 3.3. Also here, since congruent terms have the same outgoing transitions, the presented properties obviously hold also for terms $P \in \mathcal{P}_{ext}^{[w]ra}$ such that $P \equiv P'$ for some $P' \in \mathcal{P}^{[w]ra}$.

The following proposition is analogous to Proposition 3.1, the difference being that it considers $\xrightarrow{q, \hat{q}}$ transitions instead of \xrightarrow{q} transitions.

Proposition 4.1. Let $P \in \mathcal{P}^{[w]ra}$. There is no $P' \in \mathcal{P}_{ext}^{[w]ra}$, $q \in \mathcal{QN}$, $\hat{q} \in \mathcal{QN} \cup \{\varepsilon\}$ such that $P \xrightarrow{q, \hat{q}} P'$.

Proof. A direct consequence of the fact that for any $Q \equiv P$, with $P \in \mathcal{P}^{[w]ra}$, we have that, due to the laws of \equiv , all prefixes “ $\bar{x}\langle y \rangle_q$ ” and “ $\tau_{q,q'}$ ” occurring in Q are bound by a $(\nu q \rightarrow \lambda)P$ binder (rules for $(\nu q \rightarrow \lambda)P$ binders do not allow transitions $\xrightarrow{q, \hat{q}}$ labeled with the bound name q to be inferred). \square

The following proposition is analogous to Proposition 3.2. Here we additionally state that $P \longrightarrow \sigma$, with P belonging to the non-extended syntax, implies that σ is a (numerical) rate distribution (Definition 3.2). This is significant only for the version with weights and means that, for terms $P \in \mathcal{P}^{wra}$, symbolic rate distributions (that are not rate distributions) are used only during inference of stochastic reductions.

Proposition 4.2. *Let $P \in \mathcal{P}^{[w]ra}$ and $P \longrightarrow \sigma$. We have $\sigma \in RDist(\mathcal{P}_{ext}^{[w]ra} / \equiv)$ and $P \longrightarrow \sigma'$ implies $\sigma = \sigma'$.*

Proof. The proof, reported in Appendix B, is an elaboration of that of Proposition 3.2. It is, however, much more involved also due to the interplay between stochastic output binders and stochastic input binders, which, in the variant with input weights, additionally entails rate calculation by means of symbolic rate distributions during stochastic transition inference. \square

The following proposition is analogous to Proposition 3.3, the difference being that it considers terms P' that are reachable just by $P \longrightarrow \sigma$ transitions (we no longer have $P \longrightarrow P'$ transitions).

Proposition 4.3. *Let $P \in \mathcal{P}^{[w]ra}$ and $P' \in \mathcal{P}_{ext}^{[w]ra}$ such that $P \longrightarrow \sigma \wedge \exists \lambda : (\lambda, [P']_{\equiv}) \in \sigma$. There exists $P'' \in \mathcal{P}^{[w]ra}$ such that $P'' \equiv P'$.*

Proof. For any $Q \equiv P$ with $P \in \mathcal{P}^{[w]ra}$ we have: all named prefixes “ $\bar{x}\langle y \rangle_q$ ”, “ $x(y)_q$ ” and “ $\tau_{q,q'}$ ” occurring in Q are bound and every stochastic binder can bind at most a single named prefix. It is easy to verify that this property is preserved by transitions $Q \longrightarrow \sigma$ for which $\exists \lambda$ such that $(\lambda, [Q']_{\equiv}) \in \sigma$, i.e. for any such Q' the same property holds true. Performing the σ transition causes, for instance, the corresponding pair of stochastic binders not to bind any named prefix in the target state.

Any Q' satisfying the property above can be shown to be congruent to a term of P by just removing stochastic binders not binding any delay prefix and by turning bound named prefixes into Markovian action prefixes: $\bar{x}\langle y \rangle_\lambda$, τ_λ and $x(y)$ (or $x(y)_w$ for the version with weights). \square

As in the case of Markovian delays, a consequence of the above propositions and of Definition 4.8 is that, given $P \in \mathcal{P}^{[w]ra}$, we can represent the semantics of P as a transition system over states $[P']_{\equiv}$ such that $P' \in \mathcal{P}^{[w]ra}$ (with the initial state being $[P]_{\equiv}$).

Concerning the underlying Markov Chain we consider the rate $\gamma([P]_{\equiv}, [P']_{\equiv})$ of performing a reduction from state $[P]_{\equiv}$ to state $[P']_{\equiv}$ to be as we presented

in Section 3.2, that is: $\sum_{(\lambda, [P']_{\equiv}) \in \sigma} \lambda \cdot \mu_{\sigma}(\lambda, [P']_{\equiv})$ if $[P]_{\equiv} \longrightarrow \sigma$ for some σ ; 0 otherwise.

The above also provides a definition of bisimulation over states of $P \in \mathcal{P}^{[w]ra}$ by applying it to Definition 4.2 and by assuming: $\gamma([P']_{\equiv}, \tau, [P'']_{\equiv}) = \gamma([P']_{\equiv}, [P'']_{\equiv})$, for any $P', P'' \in \mathcal{P}_{ext}^{[w]ra}$ and $\gamma([P']_{\equiv}, \mu, [P'']_{\equiv}) = 0$, for any $\mu \neq \tau$ and $P', P'' \in \mathcal{P}_{ext}^{[w]ra}$. Again, obviously, w.r.t. the Markov Chain obtained by labeled operational semantics, reduction semantics of a term P yields a *closed system interpretation*, i.e. as if all (free) actions of P were bound at the top level.

4.4. Comparison Results

We now compare, for each of the two versions of the π -calculus presented in Section 4.1, our reduction semantics with classical labeled semantics, similarly to what we did for Markovian delays.

The version without weights satisfies a Harmony theorem similar to Theorem 3.1.

Theorem 4.1 (Harmony). *Let $P \in \mathcal{P}^{ra}$. We have $P \longrightarrow \sigma$ for some σ iff $P \xrightarrow{\tau_{\lambda}, id} P'$ for some λ , id and P' . Moreover, when both hold, we have that $\forall P' \in \mathcal{P}^{ra}$*

$$\sum_{(\lambda, [P']_{\equiv}) \in \sigma} \lambda \cdot \mu_{\sigma}(\lambda, [P']_{\equiv}) = \sum_{P \xrightarrow{\tau_{\lambda}, id} \equiv P'} \lambda$$

Proof. The proof, reported in Appendix B, is based on that of Proposition 4.2 in a similar way to how the proof of Theorem 3.1 relates to that of Proposition 3.2: it exploits the detected structure of the unique σ (if any) such that $P \longrightarrow \sigma$. However here we need to resort to more complex machinery, also due to the interplay between stochastic output binders and stochastic input binders. \square

Theorem 4.1 also confirms what is shown in [12], i.e. that for terms $P \in \mathcal{P}^{ra}$ with the classical approach (where the notion of apparent rate is not used) parallel is associative up to Markovian bisimulation: if in Example 4.1 we associate parallel to the left we still get two τ_2 transitions (only id labels change).

For the version with weights, instead, Theorem 4.1 does not hold: as we discussed in the Introduction this is related to the fact that, differently from our approach (where we always consider the total weight of *all* input actions synchronizable with a given output), parallel of Stochastic π -calculus [21] is not associative [12]. If we consider $Q = (\bar{x} < y >_3 . P' | x(z)_4 . P'') | x(z')_8 . P'''$, i.e. P of Example 4.2 where parallel is, instead, associated to the left, the classical approach yields (differently from P that performs a τ_1 and a τ_2 transition) two τ_3 transitions distinguished just by the id label. This provides a counterexample that shows that Theorem 4.1 does not hold for terms $P \in \mathcal{P}^{wra}$ (Q obviously has the same stochastic transition as P , we showed in Example 4.4, in that $P \equiv Q$).

5. Related Work

We here discuss other related work, besides that we already considered in the introduction.

A revisited version of original stochastic π -calculus [21] is presented in [8], where rates λ are associated to π -calculus action names and each synchronization on a name gets rate λ associated to it. From a quantitative viewpoint, thus, synchronization rates are computed similarly to our version of stochastic π -calculus with Markovian actions that considers non-quantified inputs (see Section 4). The process algebra considered in [8] makes use of binders, called “fresh name quantifiers”, that are similar to the stochastic binders in this paper (and in [2]), with the difference that they are used to directly assign rates to π -calculus action names, instead of stochastic names q (i.e. a dedicated sort of names that is used just to *symbolically represent rates*), which do not exist in [8]. Though fresh name quantifiers of [8] look similar to stochastic binders, they are used for a totally different aim. They make it possible to endow stochastic π -calculus with a novel feature: the ability to pass around actions with a bound name but with an explicit associated rate value (that is transmitted together with the action itself). We instead use rates inside binders for associating numerical values to stochastic names for the aim of distinguishing transitions. Concerning the definition of semantics, even if [8] considers a congruence relation on terms, it resorts to labeled operational semantics (and not semantics in reduction style) and does not use congruence during transition inference: this allows it to deal with rate multiplicity as in [9]. The approach of [8] cannot, therefore, be used when a semantics *in reduction style* is considered: e.g. the two Markovian *reductions* of the term in Example 4.3 cannot be *collectively* inferred by establishing a fixed structure of the parallel operators (as [8] and [9] require), in that they *individually* need a different re-arrangement of parallel. The similarity in the binder representation would maybe make it possible to apply the technique in this paper to produce a reduction semantics for the expressive algebra in [8] in an overall harmonious approach.

We would like also to mention the expressive process algebra STOcK-LAIM [10], a Markovian extension of the process algebra cKLAIM featuring mobile components interacting via multiple distributed tuple spaces. The approach used in [10] for managing the extension of cKLAIM action prefixes (acting on tuple spaces) with rates has interesting connections with the technique we presented. Being communication asynchronous, rates generated by the semantics of [10] are simply given by the cKLAIM action prefix rate, similarly to our version of stochastic π -calculus with Markovian actions that considers non-quantified inputs (where we consider outputs to be cKLAIM action prefixes that synchronize with tuple spaces). In [10], however, the problem we deal with in this paper, of managing numerical λ rates in prefixes, is avoided by assuming the STOcK-LAIM language to express Markovian durations, instead, by means of symbolic rate names, which are required to be all distinct (as a syntactic constraint on terms). Our approach makes it possible to define a semantics for a variant of the algebra in [10] with numerical λ rates inside prefixes (so that the distinguished names

considered by [10] are automatically generated).

Finally, we would like to comment on previous work concerning the issue, of preserving associativity of CCS/ π -calculus parallel when both output and input actions are quantified by rates/weights (as e.g. in stochastic π -calculus [21]). In [9] a restriction on the structure of process algebra terms is detected that makes it possible to obtain associativity of CCS parallel with a semantics in labeled style. In particular action rates/weights are managed with an elaboration of the *collective* approach of [9] that we described in the introduction. The restriction considered disallows terms with choices between outputs and inputs, e.g. term $(\bar{x}_4 + x_3) | x_2$ where the total weight of the inputs that can synchronize with \bar{x} would be erroneously considered to be 5 (our reduction semantics would instead correctly consider such total weight to be 2). Moreover the technique introduced in [9] for dealing with CCS parallel requires: some modification of traditional labeled operational semantics in that transitions obtained by synchronization do not simply become τ (they must be still labeled) and to complicate the calculation of rates in the operational rules by performing elaborate renormalizations at all syntactic levels.

6. Conclusion

In this paper we presented a technique based on stochastic names and stochastic binders to express semantics in reduction style of several Markovian extensions of π -calculus: one where additional Markovian delay prefixes are simply considered and two more complex ones where, instead, quantitative information is attached to π -calculus action prefixes, in order to generate Markovian transitions by synchronization. In particular, while in both latter extensions outputs and τ actions are endowed with rates, we considered a variant in which inputs are non-quantified and a variant in which they are, instead, endowed with weights. We showed correctness of the introduced semantic technique by comparison with the classical Markovian semantic technique of [21] in labeled style. We finally observed that, in the case of the extension using input weights, our semantics differ from Markovian semantics in the literature in that it features, for the first time, associativity for π -calculus parallel.

We now discuss possible other applications of our technique. The approach introduced in this paper, that allows us to independently identify each Markovian duration λ by simply exploiting extrusion and stochastic binders, is applicable also in the case of generally distributed durations, where the identification mechanism is needed to generate clock names (to uniquely relate start of delays with termination of delays), see [5, 3]. For example, with an approach similar to that presented in Section 3 we could define reduction semantics for a process algebra (as, e.g., IGSMP [5]) which, in addition to standard actions, considers delay prefixes f , with f being a general probability distribution of time. In particular, we could consider laws of congruence that make it possible to transform f prefixes into: a stochastic binder for a name q with associated distribution f that binds a delay start prefix followed by a delay termination prefix, both decorated with stochastic name q (clock start binders in [3] are a step in this direction).

Acknowledgements We would like to thank Luca Cardelli, Jane Hillston and Davide Sangiorgi for the fruitful comments and suggestions.

References

- [1] M. Bernardo, M. Bravetti “*Performance measure sensitive congruences for Markovian process algebras*”, in Theoretical Computer Science, 290(1):117-160, 2003
- [2] M. Bravetti “*Stochastic Semantics in the Presence of Structural Congruence: Reduction Semantics for Stochastic Pi-Calculus*”, Technical Report UBLCS-2008-16 of University of Bologna, Department of Computer Science. Available at <http://www.informatica.unibo.it/it/ricerca/technical-report/2008>
- [3] M. Bravetti, P. R. D’Argenio “*Tutte le Algebre Insieme: Concepts, Discussions and Relations of Stochastic Process Algebras with General Distributions.*”, in Validation of Stochastic Systems, LNCS 2925: 44-88, 2004
- [4] M. Bravetti, S. Gilmore, C. Guidi, M. Tribastone “*Replicating Web Services for Scalability*”, In Proc. of *Trustworthy Global Computing, Third Symposium (TGC’07)*, LNCS 4912:204-221, 2007
- [5] M. Bravetti, R. Gorrieri, “*The Theory of Interactive Generalized Semi-Markov Processes*”, in Theoretical Computer Science, 282:5-32, 2002
- [6] M. Bravetti, R. Gorrieri, “*Deciding and Axiomatizing Weak ST Bisimulation for a Process Algebra with Recursion and Action Refinement*”, in ACM Transactions on Computational Logic 3(4):465-520, ACM Press, 2002
- [7] M. Bravetti, H. Hermanns, J.-P. Katoen, “*YMCA: - Why Markov Chain Algebra?*”, Electr. Notes Theor. Comput. Sci. 162: 107-112, 2006
- [8] L. Cardelli, R. Mardare “*Stochastic Pi-calculus Revisited*”, proc. of 10th Int. Coll. on Theoretical Aspects of Computing (ICTAC 2013), LNCS 8049: 1-21, 2013
- [9] R. De Nicola, D. Latella, M. Loreti, M. Massink “*A Uniform Definition of Stochastic Process Calculi*”, ACM Computing Surveys 46(1): 5, 2013
- [10] R. De Nicola, D. Latella, M. Massink: “*Formal modeling and quantitative analysis of KLAIM-based mobile systems*”, proc. of the 2005 ACM Symposium on Applied Computing (SAC 2005), ACM, pp. 428-435, 2005
- [11] Josee Desharnais, Abbas Edalat, Prakash Panangaden: “*Bisimulation for Labelled Markov Processes*”, Information and Computation 179(2): 163-193, 2002
- [12] B. Klin, V. Sassone, “*Structural operational semantics for stochastic and weighted transition systems*”, in Information and Computation, 227:58-83, 2013
- [13] M. Giry, “*A categorical approach to probability theory*”, in Categorical Aspects of Topology and Analysis, Lecture Notes in Mathematics, 915:68-85, 1982

- [14] N. Götz, U. Herzog, M. Rettelsbach, “*TIPP - A Stochastic Process Algebra*”, proc. of 1st workshop on Process Algebras and Perf. Modeling (PAPM '93): 31-36, 1994
- [15] J.F. Groote, “*Transition system specifications with negative premises*”, in Theoretical Computer Science, 118(2):263-299, 1993
- [16] H. Hermanns, “*Interactive Markov Chains: The quest for quantified quality*”, Springer-Verlag Berlin, 2002
- [17] J. Hillston, “*A Compositional Approach to Performance Modelling*”, Cambridge University Press, 1996
- [18] E. B. Johnsen and O. Owe, “*An asynchronous communication model for distributed concurrent objects*”, in Software and System Modeling, 6(1):35-58, 2007
- [19] R. Milner, “*Communicating and mobile systems - the Pi-calculus*”, Cambridge University Press, 1999
- [20] R. Milner, J. Parrow, D. Walker “*A Calculus of Mobile Processes I and II*”, Information and Computation 100(1): 1-77, 1992
- [21] C. Priami, “*Stochastic π -Calculus*”, in Computer Journal 38(6):578-589, 1995
- [22] D. Sangiorgi, D. Walker, “*The pi-calculus: a Theory of Mobile Processes*”, Cambridge University Press, 2001
- [23] R. Segala, N. A. Lynch: “*Probabilistic Simulations for Probabilistic Processes*”, Nordic Journal of Computing 2(2): 250-273, 1995
- [24] Erik P. de Vink, Jan J. M. M. Rutten, “*Bisimulation for Probabilistic Transition Systems, A Coalgebraic Approach*”, Theoretical Computer Science 221(1-2): 271-293, 1999

Appendix A. Proofs of Section 3 Main Results

Proposition 3.2. *Let $P \in \mathcal{P}$ and $P \longrightarrow \sigma$. We have $P \longrightarrow \sigma'$ implies $\sigma = \sigma'$.*

Proof. Let n be the number of unguarded syntactic occurrences of Markovian delay prefixes inside P . Formally n is evaluated as follows: P' is obtained from P by performing a finite number of successive substitutions of constants so that all constants occur weakly guarded in P' (always possible because recursion is guarded, see Section 3); n is the number of Markovian delay prefixes that occur in P' not in the scope of a prefix operator. Since $P \longrightarrow \sigma$ we must have $n \geq 1$.

We first show, by induction on k , with $0 \leq k \leq n$ ($k = 0$ being the base case), that the following statement holds true. Let q_1, \dots, q_k be any set of distinguished names. There exists Q such that $P \equiv (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_k \rightarrow \lambda_k) Q$ (consider $P \equiv Q$ in the case $k = 0$) and $\forall i, 1 \leq h \leq k. Q \xrightarrow{q_h}$. Moreover for any such Q , we have, supposing $\forall h, 1 \leq h \leq k. Q \xrightarrow{q_h} P'_h$, that:

- Q includes a unique free unguarded syntactic occurrence of $(q_h) \forall h, 1 \leq h \leq k$ (this reduces to true for the base case $k = 0$). This assertion (and all other properties of Q stated above) derives directly from the induction hypothesis (case $k - 1$): it is shown by extruding $(\nu q_k \rightarrow \lambda_k)$ via congruence laws up to the top-level of a term Q' such that $P \equiv (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_{k-1} \rightarrow \lambda_{k-1}) Q'$. Moreover, since congruence laws preserve uniqueness of free (q) prefixes, we have that Q includes a unique free unguarded syntactic occurrence of (q_k) and unique free unguarded syntactic occurrences of $(q_h) \forall h, 1 \leq h \leq k - 1$.
- If $k < n$, for any q distinguished from $q_h \forall h, 1 \leq h \leq k$, there exist Q', λ such that $Q \equiv (\nu q \rightarrow \lambda) Q'$ and $Q' \xrightarrow{q}$, i.e. $Q \longrightarrow \sigma$. This is shown by extruding $(\nu q \rightarrow \lambda)$ via congruence laws up to the top-level of Q . Moreover, for any such Q', λ we have that Q' includes a unique free unguarded syntactic occurrence of (q) and unique free unguarded syntactic occurrences of $(q_h) \forall h, 1 \leq h \leq k$ and, supposing $\forall h, 1 \leq h \leq k. Q' \xrightarrow{q_h} P''_h$, we have that $(\nu q \rightarrow \lambda) P''_h \equiv P'_h$. This is shown as follows. Since, as already observed, Q includes a unique free unguarded syntactic occurrence of $(q_h) \forall h, 1 \leq h \leq k$ and $Q \equiv (\nu q \rightarrow \lambda) Q'$, we have the uniqueness results (due to the fact, similarly to the previous case, that congruence laws preserve uniqueness) and, consequently, that $(\nu q \rightarrow \lambda) Q' \xrightarrow{q_h} (\nu q \rightarrow \lambda) P''_h$ with $(\nu q \rightarrow \lambda) P''_h \equiv P'_h$.
- If, instead, $k = n$ there is no Q', q, λ such that $Q \equiv (\nu q \rightarrow \lambda) Q'$ and $Q' \xrightarrow{q}$, i.e. $\nexists \sigma: Q \longrightarrow \sigma$. This is because, due again to the fact that congruence laws preserve uniqueness, the only occurrences of (q) prefixes in Q are the unique free unguarded syntactic occurrences of $(q_h) \forall h, 1 \leq h \leq n$ and in Q there is no λ such that Q includes an unguarded syntactic occurrence of (λ) .

Considering case $k = n$ of the above, we have that, given distinguished names q_1, \dots, q_n , there exists P' such that $P \equiv (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) P'$ and

$\forall i, 1 \leq h \leq n. P' \xrightarrow{q_h}$. Moreover for any such P' we have that P' includes a unique free unguarded syntactic occurrence of $(q_i) \forall i, 1 \leq i \leq n$.

In the following we will show, by induction on the length of inference chains, that inferring a σ transition from P (where q_1 is the name of the binder used to infer σ in the last rule used in the inference chain and so on and so forth in depth until q_n where the chain of binder applications begins with the *(Sto2)* rule) entails singling out one such term P' and that, supposing $\forall h, 1 \leq h \leq n. P' \xrightarrow{q_h} P'_h$, the inferred σ is such that

$$\sigma = \{ \{ (\lambda_1, [(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) P'_1]_{\equiv}) \} \cup \dots \cup \{ (\lambda_n, [(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) P'_n]_{\equiv}) \} \}$$

Notice that this multiset does not depend on the choice of the P'_h such that $P' \xrightarrow{q_h} P'_h$ in that, since P' includes a unique free unguarded syntactic occurrence of (q_h) , possible terms P'_h are all related by \equiv (and, in general, $P \equiv Q$ implies $(\nu q \rightarrow \lambda)P \equiv (\nu q \rightarrow \lambda)Q$ for any P, Q, q and λ).

From this fact we can derive that the σ inferred is the same for any possible inference chain of a σ transition from P , i.e. that no matter the order in which delay prefixes are bound, the same rate distribution is obtained. This is because, considering σ' derived with another inference chain, such an inference would determine a P'' such that $P \equiv (\nu q'_1 \rightarrow \lambda'_1) \dots (\nu q'_n \rightarrow \lambda'_n) P''$ and $\forall i, 1 \leq i \leq n. P'' \xrightarrow{q'_i}$. It is immediate to verify that there exists a permutation k_i , with $1 \leq i \leq n$, such that, by alpha-conversion we can write $(\nu q'_1 \rightarrow \lambda'_1) \dots (\nu q'_n \rightarrow \lambda'_n) P''$ as $(\nu q_{k_1} \rightarrow \lambda_{k_1}) \dots (\nu q_{k_n} \rightarrow \lambda_{k_n}) P'''$ where for P''' it holds $P''' \equiv P'$. Thus $\sigma' = \{ \{ (\lambda_{k_1}, [(\nu q_{k_1} \rightarrow \lambda_{k_1}) \dots (\nu q_{k_n} \rightarrow \lambda_{k_n}) P'_1]_{\equiv}) \} \cup \dots \cup \{ (\lambda_{k_n}, [(\nu q_{k_1} \rightarrow \lambda_{k_1}) \dots (\nu q_{k_n} \rightarrow \lambda_{k_n}) P'_n]_{\equiv}) \} \}$, where $\forall i, 1 \leq i \leq n$ we have $P'_i \equiv P'_{k_i}$ because P' includes a unique free unguarded syntactic occurrence of $(q_i) \forall i, 1 \leq i \leq n$. $\sigma' = \sigma$ by reordering of the binders, of the members of the multiset union and from the fact that $P \equiv Q$ implies $(\nu q \rightarrow \lambda)P \equiv (\nu q \rightarrow \lambda)Q$ for any P, Q, q and λ .

Let us now analyze the structure of the inference chain of a σ transition from P by considering an inductive argument (starting from $k = 0$). Given k such that $0 \leq k \leq n - 1$, the structure of any inference sequence of any σ transition of Q such that $P \equiv (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_k \rightarrow \lambda_k) Q$ (consider $P \equiv Q$ in the case $k = 0$) and $\forall h, 1 \leq h \leq k. Q \xrightarrow{q_h} P'_h$ must be as follows. Apart from applications of the rule *(StoCong)*, that can be used to get, from $Q \rightarrow \sigma, (\nu q_{k+1} \rightarrow \lambda_{k+1}) Q' \rightarrow \sigma$, with Q' such that $Q \equiv (\nu q_{k+1} \rightarrow \lambda_{k+1}) Q'$ and $Q' \xrightarrow{q_{k+1}} P'_{k+1}$ (we have already shown such Q' to always exist for $k \leq n - 1$) the last rule used must be:

- *(Sto2)* if $k = n - 1$, yielding directly $(\nu q_n \rightarrow \lambda_n) Q' \rightarrow \sigma$. This is because, as we have shown already, $\nexists \sigma' : Q' \rightarrow \sigma'$. The inference sequence is therefore terminated (as far as stochastic binder application is concerned).
- *(Sto1)* if $k < n - 1$, yielding $(\nu q_{k+1} \rightarrow \lambda_{k+1}) Q' \rightarrow \sigma$. This is because, as we have shown already, $\exists \sigma' : Q' \rightarrow \sigma'$. The inference sequence is therefore continued by the inference of a σ' transition of term Q' .

Hence, by repeatedly applying the statement above (n times starting from $k = 0$) we have that, for any σ transition of P there exist Q_1, \dots, Q_n , with $P \equiv (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_k \rightarrow \lambda_k) Q_k$ for $1 \leq k \leq n$, such that the inference sequence of the σ transition must include $n - 1$ uses of the *(Sto1)* rule yielding $(\nu q_1 \rightarrow \lambda_1) Q_1 \longrightarrow \sigma_1, \dots, (\nu q_{n-1} \rightarrow \lambda_{n-1}) Q_{n-1} \longrightarrow \sigma_{n-1}$ transitions, with $\sigma_1 = \sigma$, followed by one use of the *(Sto2)* rule yielding $(\nu q_n \rightarrow \lambda_n) Q_n \longrightarrow \sigma_n$ transition. We have, supposing $\forall h, 1 \leq h \leq n. Q_n \xrightarrow{q_h} P'_h, \sigma_k = \{ \{ (\lambda_k, [(\nu q_k \rightarrow \lambda_k) \dots (\nu q_n \rightarrow \lambda_n) P'_k]_{\equiv}) \} \} \cup \dots \cup \{ (\lambda_n, [(\nu q_k \rightarrow \lambda_k) \dots (\nu q_n \rightarrow \lambda_n) P'_n]_{\equiv}) \} \}$. This is shown by induction on the length of the inference sequence (in terms of number of applications of *(Sto1)*/*(Sto2)* rules). The base case is for $k = n$ (application of the *(Sto2)* rule); then we show it holds for k supposing that it holds for $k + 1$ (application of the *(Sto1)* rule). Notice that the induction is performed in the reversed direction w.r.t. above. In the inductive case we exploit the fact, showed before, that, given P''_h such that $Q_{k+1} \xrightarrow{q_h} P''_h$, we have, for P'_h with $Q_k \xrightarrow{q_h} P''_h$, that $(\nu q_{k+1} \rightarrow \lambda_{k+1}) P''_h \equiv P''_h$. \square

Theorem 3.1 (Harmony). *Let $P \in \mathcal{P}$. We have:*

- $P \longrightarrow P' \text{ iff } P \xrightarrow{\tau} \equiv P'$
- $P \longrightarrow \sigma \text{ for some } \sigma \text{ iff } P \xrightarrow{\lambda, id} P' \text{ for some } \lambda, id \text{ and } P'. \text{ Moreover, when both hold, we have that } \forall P' \in \mathcal{P}$

$$\sum_{(\lambda, [P']_{\equiv}) \in \sigma} \lambda \cdot \mu_{\sigma}(\lambda, [P']_{\equiv}) = \sum_{P \xrightarrow{\lambda, id} \equiv P'} \lambda$$

Proof. The proof of the first item is standard in that reduction and τ transitions are derived via standard rules (they are not affected by the presence of Markovian delays), see, e.g. the Harmony Lemma in [22]. In the following we present the proof of the second item.

Let n be the number of unguarded syntactic occurrences of Markovian delay prefixes inside P (see proof of Proposition 3.2 for a formal definition of n). We have that $P \longrightarrow \sigma$ for some σ iff $n > 0$ (the “if” part is obvious, see proof of Proposition 3.2 for the “only if”). Moreover we also obviously have that $P \xrightarrow{\lambda, id} P'$ for some λ, id and P' iff $n > 0$.

Assuming $n > 0$, we consider term Q obtained from P by performing a finite number of successive substitutions of constants so that all constants occur weakly guarded in Q (always possible because recursion is guarded, see Section 3). Notice that since $P \equiv Q$ we have $Q \longrightarrow \sigma$. Moreover it is immediate to verify that for each Q' such that $Q \xrightarrow{\lambda, id} Q'$ there exists P' , with $P' \equiv Q'$, such that $P \xrightarrow{\lambda, id} P'$; and vice versa.

Given distinguished names q_1, \dots, q_n , let Q' and rates $\lambda_i, 1 \leq i \leq n$, with Q' including a unique free unguarded syntactic occurrence of each (q_i) $\forall i, 1 \leq i \leq n$, be such that $Q' \{ \lambda_i / q_i \mid 1 \leq i \leq n \} = Q$. We have that $Q \equiv (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) Q'$ (by extruding stochastic binders up to the top

level). Moreover $\forall h, 1 \leq h \leq n$, let Q_h be $Q' \{ \lambda_i / q_i \mid 1 \leq i \leq n \wedge i \neq h \}$. We have that $Q \equiv (\nu q_h \rightarrow \lambda_h) Q_h$ (by extruding the stochastic binder q_h up to the top level).

We now resort to classical correspondence of transitions between reduction and labeled semantics. We have: $Q_h \xrightarrow{q_h} Q'_h$ iff $Q \xrightarrow{\lambda_h, id_h} Q''_h$ with $Q''_h \equiv Q'_h$. This is proved in a completely analogous way as for the Harmony Lemma in [22], since, both unbound (q_h) prefixes for the reduction semantics in this paper and (λ) for the labeled semantics in this paper have the same behavior as τ prefixes for standard reduction and labeled semantics (they are just “consumed” in the target state with no other effects).

Considering Q' we have that $Q' \xrightarrow{q_h} Q'''_h$ with $Q'_h \equiv (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) Q''_h$ (see proof of Proposition 3.2). Hence $\sigma = \{ \{ (\lambda_1, [(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) Q''_1]_{\equiv}) \} \cup \dots \cup \{ (\lambda_n, [(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) Q''_n]_{\equiv}) \} \}$. Thus, considering the set I of the indexes h , with $1 \leq h \leq n$, such that $[(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) Q''_h]_{\equiv} = [P']_{\equiv}$ we have $\sum_{(\lambda, [P']_{\equiv}) \in \sigma} \lambda \cdot \mu_{\sigma}(\lambda, [P']_{\equiv}) = \sum_{h \in I} \lambda_h$. Moreover I is also the set of indexes h such that $Q''_h \equiv P'$, because $Q''_h \equiv (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) Q''_h$. Therefore, since $h \neq h'$ implies $id_h \neq id_{h'}$, we also have $\sum_P \xrightarrow{\lambda, id} \equiv P' \lambda = \sum_Q \xrightarrow{\lambda, id} \equiv P' \lambda = \sum_{h \in I} \lambda_h$. \square

Appendix B. Proofs of Section 4 Main Results

Proposition 4.2. *Let $P \in \mathcal{P}^{[w]ra}$ and $P \longrightarrow \sigma$. We have $\sigma \in RDist(\mathcal{P}_{ext}^{[w]ra} / \equiv)$ and $P \longrightarrow \sigma'$ implies $\sigma = \sigma'$.*

Proof. We start by considering the version without weights.

Let n_{all} be the number of unguarded syntactic occurrences of output and τ prefixes (τ are seen as both outputs and inputs) inside P and m be the number of unguarded syntactic occurrences of input and τ prefixes (taken again into account) inside P . Formally n_{all} and m are evaluated as follows: P' is obtained from P by performing a finite number of successive substitutions of constants so that all constants occur weakly guarded in P' (always possible because recursion is guarded, see Section 3); n_{all} and m are the number of the prefixes described above that occur in P' not in the scope of a prefix operator. Let us now suppose:

- The n_{all} syntactic occurrences of output and τ prefixes to be uniquely identified via indexes h , with $1 \leq h \leq n_{all}$ and let λ_h be the rate associated with the occurrence identified with h ($\bar{x} \langle y \rangle_{\lambda_h}$, for some x, y , or τ_{λ_h}).
- The m syntactic occurrences of $x(y)$ input (for some x, y) and τ prefixes to be uniquely identified via indexes h' (τ prefixes are, thus, identified in two ways depending on if we are considering them when dealing with output names or when dealing with input names, see below), with $1 \leq h' \leq m$.
- The assigned indexes above are such that immediately synchronizable outputs and τ are listed at the beginning: they are indexed by the first n indexes. That is, formally, consider $(\nu q'_1) \dots (\nu q'_m)(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_{n_{all}} \rightarrow$

$\lambda_{n_{all}})P'' \equiv P'$, where P'' syntactically differs from P' just for the fact that each unguarded syntactic occurrence of $\bar{x}<y>_{\lambda_h}$ (for some x, y) identified by h , has been replaced by $\bar{x}<y>_{q_h}$, for some name q_h ; each unguarded syntactic occurrence of $x(y)$ (for some x, y) identified by h' has been replaced by $x(y)_{q'_{h'}}$, for some name $q'_{h'}$; and each unguarded syntactic occurrence of τ_{λ_h} identified by h (as output) and h' (as input) has been replaced by $\tau_{q_h, q'_{h'}}$, for some names q_h and $q'_{h'}$. Indexes are assigned so

that there exists $n \leq n_{all}$ with $\{h \mid 1 \leq h \leq n\} = \{h \mid \exists h'. P'' \xrightarrow{q_h, q'_{h'}}\}$.

Since $P \rightarrow \sigma$ we must have $n \geq 1$ and $m \geq 1$.

In the following we will assume function H_h to map each output/ τ index h , with $1 \leq h \leq n$, into the non-empty set of input/ τ indexes $\{h' \mid P'' \xrightarrow{q_h, q'_{h'}}\}$ (intuitively the index of an output is mapped into the set of the indexes of the inputs it can synchronize with and the index of a syntactic occurrence of τ , seen as an output, is mapped into a singleton including the index of the same syntactic occurrence of τ , seen as an input).

We first show that the following statement holds true, by induction on k , with $0 \leq k \leq n$ ($k = 0$ being the base case). Let q_1, \dots, q_k be any set of distinguished names. Let also $q'_{k_1}, \dots, q'_{k_{k'}}$ be any set of k' distinguished names, where k_i is injective and $\cup_{1 \leq i \leq k} H_i \subseteq \{k_i \mid 1 \leq i \leq k'\}$. There exists Q such that $P \equiv (\nu q'_{k_1}) \dots (\nu q'_{k_{k'}})(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_k \rightarrow \lambda_k)Q$ (consider $P \equiv Q$ in the case $k = 0$) and $\forall h, 1 \leq h \leq k. \forall h' \in H_h. Q \xrightarrow{q_h, q'_{h'}}$. Moreover for any such Q , we have, supposing $\forall h, 1 \leq h \leq k. \forall h' \in H_h. Q \xrightarrow{q_h, q'_{h'}} P'_{h, h'}$, that:

- $\forall h, 1 \leq h \leq k$ we have that (this reduces to true for the base case $k = 0$):
 - either Q includes a unique free unguarded syntactic occurrence of $\tau_{q_h, q'_{h'}}$, with $H_h = \{h'\}$;
 - or there exists x such that Q includes a unique free unguarded syntactic occurrence of $\bar{x}<y>_{q_h}$ (for some y) and, for each $h' \in H_h$, a unique free unguarded syntactic occurrence of $x(y)_{q'_{h'}}$ (for some y).

This assertion (and all other properties of Q stated above) derives directly from the induction hypothesis (case $k - 1$): it is shown by extruding $(\nu q_k \rightarrow \lambda_k)$ and all $(\nu q'_{h'})$, with $h' \in H_k - \{k_i \mid 1 \leq i \leq k'\}$, via congruence laws up to the top-level of a term Q' such that $P \equiv (\nu q'_{k_1}) \dots (\nu q'_{k_{k'}})(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_{k-1} \rightarrow \lambda_{k-1})Q'$ with the above properties. In particular, since congruence laws preserve uniqueness of prefixes, we have: by performing such an extrusion, that Q includes the required unique syntactic occurrences of prefixes for $h = k$; by induction, that Q includes the required unique syntactic occurrences of prefixes for h such that $1 \leq h \leq k - 1$.

- If $k < n$, there exist Q', λ , injective k'_i, k'', q distinguished from q_h $\forall h, 1 \leq h \leq k$, and $\{q''_{k'_i} \mid 1 \leq i \leq k''\} \cap \{q'_{k'_i} \mid 1 \leq i \leq k'\} = \emptyset$ such that

$Q \equiv (\nu q''_{k'_1}) \dots (\nu q''_{k''_{k''}})(\nu q \rightarrow \lambda)Q'$, and $\varepsilon \notin \{\widehat{q} \mid Q' \xrightarrow{q, \widehat{q}}\} \neq \emptyset$, i.e. $Q \longrightarrow \sigma$. This is shown by considering j such that $k < j \leq n$ and k'_i, k'' such that $H_j - \{k_i \mid 1 \leq i \leq k'\} \subseteq \{k'_i \mid 1 \leq i \leq k''\}$ and by extruding $(\nu q \rightarrow \lambda)$ and $(\nu q''_{k'_1}) \dots (\nu q''_{k''_{k''}})$ via congruence laws up to the top-level of Q . Moreover, for any such $Q', \lambda, k'_i, k'', q$ and $\{q''_{k'_i} \mid 1 \leq i \leq k''\}$ we have, supposing $\forall h, 1 \leq h \leq k. \forall h' \in H_h. Q' \xrightarrow{q_h, q'_{h'}} P''_{h, h'}$, that $(\nu q''_{k'_1}) \dots (\nu q''_{k''_{k''}})(\nu q \rightarrow \lambda)P''_{h, h'} \equiv P'_{h, h'}$. This is shown as follows. Due to uniqueness of prefixes shown in the previous item (and due to the fact, similarly to the previous item, that congruence laws preserve uniqueness) we have that $(\nu q''_{k'_1}) \dots (\nu q''_{k''_{k''}})(\nu q \rightarrow \lambda)Q' \xrightarrow{q_h, q'_{h'}} (\nu q''_{k'_1}) \dots (\nu q''_{k''_{k''}})(\nu q \rightarrow \lambda)P''_{h, h'}$ with $(\nu q''_{k'_1}) \dots (\nu q''_{k''_{k''}})(\nu q \rightarrow \lambda)P''_{h, h'} \equiv P'_{h, h'}$.

- If, instead, $k = n$ there is no Q', q, λ , injective k'_i and k'' such that $Q \equiv (\nu q''_{k'_1}) \dots (\nu q''_{k''_{k''}})(\nu q \rightarrow \lambda)Q'$, and $\varepsilon \notin \{\widehat{q} \mid Q' \xrightarrow{q, \widehat{q}}\} \neq \emptyset$, i.e. $Q \longrightarrow \sigma$. This is because, due again to the fact that congruence laws preserve uniqueness, the only occurrences of $\bar{x} \langle y \rangle_q$ (for some x, y) prefixes in Q are the unique free unguarded syntactic occurrences of $\bar{x} \langle y \rangle_{q_h} \forall h, 1 \leq h \leq n$ and in Q there is no λ such that Q includes an unguarded syntactic occurrence of $\bar{x} \langle y \rangle_\lambda$ (for some x, y).

Considering the case $k = n$ of the above, we have that, given distinguished names q'_1, \dots, q'_m and q_1, \dots, q_n , there exists P' such that $P \equiv (\nu q'_1) \dots (\nu q'_m)(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n)P'$ and $\forall h, 1 \leq h \leq n. \forall h' \in H_h. Q \xrightarrow{q_h, q'_{h'}} \cdot$. Moreover for any such P' we have that each name $q_i \forall i, 1 \leq i \leq n$ and $q'_j \forall j, 1 \leq j \leq m$ has a unique free occurrence inside an unguarded prefix of P' .

In the following we will show, by induction on the length of inference chains, that inferring a σ transition from P (where q_1 is the name of the binder used to infer σ in the last rule used in the inference chain and so on and so forth in depth until q_n where the chain of binder applications begins with the *(StoOut2)* rule) entails singling out one such term P' and that, supposing $\forall h, 1 \leq h \leq$

$n. \forall h' \in H_h. P' \xrightarrow{q_h, q'_{h'}} P'_{h, h'}$, the inferred σ is such that $\sigma =$
 $\{ \{ \lambda_1, [(\nu q'_1) \dots (\nu q'_m)(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n)P'_{1, h}] \equiv \} \mid h \in H_1 \} \cup \dots \cup$
 $\{ \{ \lambda_n, [(\nu q'_1) \dots (\nu q'_m)(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n)P'_{n, h}] \equiv \} \mid h \in H_n \}.$
 Notice that this multiset does not depend on the choice of the $P'_{h, h'}$ such that

$P' \xrightarrow{q_h, q'_{h'}} P'_{h, h'}$ in that, since P' includes a unique free occurrence of q_h and $q'_{h'}$ inside unguarded prefixes of P' (i.e. either an unguarded occurrence of a $\tau_{q_h, q'_{h'}}$ prefix or of an output prefix with name q_h and an input prefix with name $q'_{h'}$), possible terms $P'_{h, h'}$ are all related by \equiv (and, in general, $P \equiv Q$ implies $(\nu q \rightarrow \lambda)P \equiv (\nu q \rightarrow \lambda)Q$ and $(\nu q)P \equiv (\nu q)Q$ for any P, Q, q and λ).

From this fact we can derive that the σ inferred is the same for any possible inference chain of a σ transition from P , i.e. that no matter the order in which

delay prefixes are bound, the same rate distribution is obtained. This is because, considering σ' derived with another inference chain, such an inference would determine a P'' such that $P \equiv (\nu q_1''') \dots (\nu q_m''') (\nu q_1'' \rightarrow \lambda_1) \dots (\nu q_n'' \rightarrow \lambda_n) P''$ and $\forall h, 1 \leq h \leq n. \forall h' \in H_h. P'' \xrightarrow{q_h'', q_{h'}'''} .$ It is immediate to verify that there exists a permutation k'_i , with $1 \leq i \leq n$, and permutation a k''_i , with $1 \leq i \leq m$ such that, by alpha-conversion we can write $(\nu q_1''') \dots (\nu q_m''') (\nu q_1'' \rightarrow \lambda_1) \dots (\nu q_n'' \rightarrow \lambda_n) P''$ as $(\nu q_{k'_1}'') \dots (\nu q_{k'_n}'') (\nu q_{k'_1} \rightarrow \lambda_{k'_1}) \dots (\nu q_{k'_n} \rightarrow \lambda_{k'_n}) P'''$ where for P''' it holds $P''' \equiv P'$. Thus $\sigma' = \{(\lambda_{k_1}, [(\nu q_{k'_1}'') \dots (\nu q_{k'_n}'') (\nu q_{k'_1} \rightarrow \lambda_{k'_1}) \dots (\nu q_{k'_n} \rightarrow \lambda_{k'_n}) P_{1,h}'']) \mid h \in H_{k_1}\} \cup \dots \cup \{(\lambda_{k_n}, [(\nu q_{k'_1}'') \dots (\nu q_{k'_n}'') (\nu q_{k'_1} \rightarrow \lambda_{k'_1}) \dots (\nu q_{k'_n} \rightarrow \lambda_{k'_n}) P_{n,h}'']) \mid h \in H_{k_n}\}$, where $\forall i, 1 \leq i \leq n$ and $\forall h \in H_i$ we have $P_{i,h}'' \equiv P'_{k_i, h}$ because each name q_i $\forall i, 1 \leq i \leq n$ and q'_j $\forall j, 1 \leq j \leq m$ has a unique free occurrence inside an ungarded prefix of P' . $\sigma' = \sigma$ by reordering of the binders, of the members of the multiset union and from the fact that $P \equiv Q$ implies $(\nu q \rightarrow \lambda)P \equiv (\nu q \rightarrow \lambda)Q$ and $(\nu q)P \equiv (\nu q)Q$ for any P, Q, q and λ .

Let us now analyze the structure of the inference chain of a σ transition from P by considering an inductive argument (starting from $k = 0$). Given k such that $0 \leq k \leq n - 1$ and given k' and injective k_i such that $\cup_{1 \leq i \leq k} H_i \subseteq \{k_i \mid 1 \leq i \leq k'\}$, the structure of any inference sequence of any σ transition of Q such that $P \equiv (\nu q_{k_1}') \dots (\nu q_{k_{k'}}') (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_k \rightarrow \lambda_k) Q$ (consider $P \equiv Q$ in the case

$k = 0$) and $\forall h, 1 \leq h \leq k. \forall h' \in H_h. Q \xrightarrow{q_h, q_{h'}'} .$ must be as follows. Apart from applications of the rule (*StoCong*), the last rule used to obtain $Q \rightarrow \sigma$ must be one of (*StoOut1*), (*StoOut2*) or (*StoIn*). In particular applications of the (*StoIn*) rule for at least all binders $(\nu q'_i)$ such that $i \in H_{k+1} - \{k_i \mid 1 \leq i \leq k'\}$ must all follow the application of the last (*StoOut1*)/(*StoOut2*) rule in that for inferring $(\nu q_{k+1} \rightarrow \lambda_{k+1})Q' \rightarrow \sigma$ its requirement $\varepsilon \notin \{\hat{q} \mid Q' \xrightarrow{q, \hat{q}}\} \neq \emptyset$ must be satisfied (in a term $(\nu q_{k'_1}') \dots (\nu q_{k_{k'}}') (\nu q_{k+1} \rightarrow \lambda_{k+1})Q' \equiv Q$ where $\{q'_{k'_i} \mid 1 \leq i \leq k''\}$ does not include all indexes $H_{k+1} - \{k_i \mid 1 \leq i \leq k'\}$ we have that, given an excluded index i , Q' may include an input $x(y)_{q'_i}$ prefix, for some x, y , or a τ_{q, q'_i} , for some q , only if its bound by a $(\nu q'_i)$, hence the requirement above is violated). In particular, the rule used to obtain $(\nu q_{k+1} \rightarrow \lambda_{k+1})Q' \rightarrow \sigma$ must be:

- (*StoOut2*) if $k = n - 1$, yielding directly $(\nu q_n \rightarrow \lambda_n)Q' \rightarrow \sigma$. This is because, as we have shown already, $\nexists \sigma' : Q' \rightarrow \sigma'$ (and $\varepsilon \notin \{\hat{q} \mid Q' \xrightarrow{q, \hat{q}}\} \neq \emptyset$).

The inference sequence is therefore terminated (as far as stochastic binder application is concerned).

- (*StoOut1*) if $k < n - 1$, yielding $(\nu q_{k+1} \rightarrow \lambda_{k+1})Q' \rightarrow \sigma$. This is because, as we have shown already, $\exists \sigma' : Q' \rightarrow \sigma'$ (and $\varepsilon \notin \{\hat{q} \mid Q' \xrightarrow{q, \hat{q}}\} \neq \emptyset$). The inference sequence is therefore preceded by the inference of a σ' transition of term Q' .

Hence, by repeatedly applying the statement above (n times starting from $k = 0$) we have that, for any σ transition of P there exist Q_1, \dots, Q_n , such that:

- for $1 \leq k \leq n$ there exists injective k_i and k' such that $P \equiv (\nu q'_{k_1}) \dots (\nu q'_{k_{k'}}) (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_k \rightarrow \lambda_k) Q_k$, with the above described properties
- the inference sequence of the σ transition must include $n - 1$ uses of the $(StoOut1)$ rule yielding $(\nu q_1 \rightarrow \lambda_1) Q_1 \longrightarrow \sigma_1, \dots, (\nu q_{n-1} \rightarrow \lambda_{n-1}) Q_{n-1} \longrightarrow \sigma_{n-1}$ transitions followed by one use of the $(StoOut2)$ rule yielding a $(\nu q_n \rightarrow \lambda_n) Q_n \longrightarrow \sigma_n$ transition.

We have, supposing $\forall h, 1 \leq h \leq n. \forall h' \in H_h. Q_n \xrightarrow{q_h, q'_{h'}} P'_{h, h'}$, that $\sigma_k = \{ \{ (\lambda_k, [(\nu q'_{k_1}) \dots (\nu q'_{k_{k'}}) (\nu q_k \rightarrow \lambda_k) \dots (\nu q_n \rightarrow \lambda_n) P'_{k, h}]) \mid h \in H_k \} \cup \dots \cup \{ (\lambda_n, [(\nu q'_{k_1}) \dots (\nu q'_{k_{k'}}) (\nu q_k \rightarrow \lambda_k) \dots (\nu q_n \rightarrow \lambda_n) P'_{n, h}]) \mid h \in H_n \} \}$ for some injective k_i and k' such that $\{k_i \mid 1 \leq i \leq k'\} \cap (\cup_{1 \leq i \leq k} H_i) = \emptyset$.

This is shown by induction on the length of the inference sequence (in terms of number of applications of $(StoOut1)/(StoOut2)$ rules). The base case is for $k = n$ (application of the $(StoOut2)$ rule); then we show it holds for k supposing that it holds for $k + 1$ (application of the $(StoOut1)$ rule). Notice that the induction is performed in the reversed direction w.r.t. above. In the inductive case we exploit the fact, we showed before, that, assuming $Q_k \equiv (\nu q'_{k_1}) \dots (\nu q'_{k_{k'}}) (\nu q_{k+1} \rightarrow \lambda_{k+1}) Q_{k+1}$, we have: given $P'''_{h, h'}$ such that $Q_{k+1} \xrightarrow{q_h, q'_{h'}} P'''_{h, h'}$, we have, for $P''_{h, h'}$ with $Q_k \xrightarrow{q_h, q'_{h'}} P''_{h, h'}$, that $(\nu q'_{k_1}) \dots (\nu q'_{k_{k'}}) (\nu q_{k+1} \rightarrow \lambda_{k+1}) P'''_{h, h'} \equiv P''_{h, h'}$.

We can, therefore, conclude, concerning multiset σ (and analogously for σ'), that $\sigma = \{ \{ (\lambda_1, [(\nu q'_1) \dots (\nu q'_m) (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) P'_{1, h}]) \mid h \in H_1 \} \cup \dots \cup \{ (\lambda_n, [(\nu q'_1) \dots (\nu q'_m) (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) P'_{n, h}]) \mid h \in H_n \} \}$.

We now consider the version with weights.

Let w_h , for $1 \leq h \leq m$, be the weight associated with the occurrence of $x(y)$ input (for some x, y) prefix identified with h . The most important modifications of the proof that involve weights concern multiset σ_k above derived during inference of a stochastic transition from P and, correspondingly, multiset σ (and σ') such that $P \longrightarrow \sigma$ ($P \longrightarrow \sigma'$). In the rest of the proof it is sufficient, whenever an input binder is considered, to include in the binder the corresponding weight.

Concerning multiset σ_k , it becomes $\sigma_k = \{ \{ (E'_{k, h}, [(\nu q'_{k_1} \rightarrow w_{k_1}) \dots (\nu q'_{k_{k'}} \rightarrow w_{k_{k'}}) (\nu q_k \rightarrow \lambda_k) \dots (\nu q_n \rightarrow \lambda_n) P'_{k, h}]) \mid h \in H_k \} \cup \dots \cup \{ (E'_{n, h}, [(\nu q'_{k_1} \rightarrow w_{k_1}) \dots (\nu q'_{k_{k'}} \rightarrow w_{k_{k'}}) (\nu q_k \rightarrow \lambda_k) \dots (\nu q_n \rightarrow \lambda_n) P'_{n, h}]) \mid h \in H_n \} \}$ where $\forall h, 1 \leq h \leq n$ and $\forall h' \in H_h$ we have: $E'_{h, h'} = E_{h, h'}[w_{k_1}/q'_{k_1}] \dots [w_{k_{k'}}/q'_{k_{k'}}]$ and, in turn, $E_{h, h'} = (\lambda_h \cdot q_{h'}) / \sum_{i \in I_h} q_i$ with $\{q_i \mid i \in I_h\} = \{q_{h''} \mid h'' \in H_h\}$. Notice that, since σ_k is obtained by a $(StoOut1)/(StoOut2)$ rule, we always have $E'_{k, h} = E_{k, h}$ for all $h \in H_k$.

Finally, concerning multiset σ (and analogously for σ'), it becomes $\sigma = \{ (\lambda_1, h, [(\nu q'_1 \rightarrow w_1) \dots (\nu q'_m \rightarrow w_m) (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) P'_{1, h}]) \mid h \in H_1 \}$

$\cup \dots \cup$
 $\{(\lambda_{n,h}, [(\nu q'_1 \rightarrow w_1) \dots (\nu q'_m \rightarrow w_m)(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n)P'_{n,h}]) \mid h \in H_n\}$
 where $\forall h, 1 \leq h \leq n$ and $\forall h' \in H_h$ we have: $\lambda_{h,h'} = (\lambda_h \cdot w_{h'}) / \sum_{h'' \in H_h} w_{h''}$. \square

Theorem 4.1 (Harmony). *Let $P \in \mathcal{P}^{ra}$. We have $P \rightarrow \sigma$ for some σ iff $P \xrightarrow{\tau_\lambda, id} P'$ for some λ , id and P' . Moreover, when both hold, we have that $\forall P' \in \mathcal{P}^{ra}$*

$$\sum_{(\lambda, [P'] \equiv \sigma)} \lambda \cdot \mu_\sigma(\lambda, [P'] \equiv \sigma) = \sum_{P \xrightarrow{\tau_\lambda, id} P'} \lambda$$

Proof. Let n and m and H_h be defined as described at the beginning of the proof of Proposition 4.2. We have that $P \rightarrow \sigma$ for some σ iff $n \geq 1$ and $m \geq 1$ (the “if” part is obvious, see proof of Proposition 4.2 for the “only if”). Moreover we also obviously have that $P \xrightarrow{\tau_\lambda, id} P'$ for some λ , id and P' iff $n \geq 1$ and $m \geq 1$.

Assuming $n \geq 1$ and $m \geq 1$, we consider term Q obtained from P by performing a finite number of successive substitutions of constants so that all constants occur weakly guarded in Q (always possible because recursion is guarded, see Section 3). Notice that since $P \equiv Q$ we have $Q \rightarrow \sigma$. Moreover it is immediate to verify that for each Q' such that $Q \xrightarrow{\tau_\lambda, id} Q'$ there exists P' , with $P' \equiv Q'$, such that $P \xrightarrow{\tau_\lambda, id} P'$; and vice versa.

Let q_1, \dots, q_n and q'_1, \dots, q'_m be distinguished names. Moreover, let Q' and rates λ_i , $1 \leq i \leq n$, with Q' including a unique free occurrence of each q_i (with $1 \leq i \leq n$) and q'_i (with $1 \leq i \leq m$) inside an unguarded prefix, be such that $Q' \{ \lambda_i / q_i, \widehat{q}_j \mid 1 \leq i \leq n \wedge j \notin J \} \{ \varepsilon / q'_j \mid j \in J \} = Q$, with $\{ \widehat{q}_j \mid j \in J \}$ being the set of names included in input prefixes. We have that $Q \equiv (\nu q'_1) \dots (\nu q'_m) (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) Q'$ (by extruding stochastic binders up to the top level). Moreover $\forall h. 1 \leq h \leq n$ and $\forall h'. h' \in H_h$, let $Q_{h,h'}$ be $Q' \{ \lambda_i / q_i, \widehat{q}_j \mid 1 \leq i \leq n \wedge i \neq h \wedge j \notin J \} \{ \varepsilon / q'_j \mid j \in J \wedge j \neq h' \}$. We have that $Q \equiv (\nu q'_{h'}) (\nu q_h \rightarrow \lambda_h) Q_{h,h'}$ (by extruding the stochastic binders q_h and $q'_{h'}$ up to the top level).

We now resort to a classical correspondence of transitions between reduction and labeled semantics. We have: $Q_{h,h'} \xrightarrow{q_h, q_{h'}} Q'_{h,h'}$ iff $Q \xrightarrow{\tau_{\lambda_h}, id_{h,h'}} Q''_{h,h'}$ with $Q''_{h,h'} \equiv Q'_{h,h'}$.

This is proved in a completely analogous way as for the Harmony Lemma in [22], since, both prefixes including unbound names for the reduction semantics in this paper and Markovian action prefixes for the labeled semantics in this paper have the same behavior as prefixes for standard reduction and labeled semantics (they are just “consumed” in the target state with no other effects).

Considering Q' we have that $Q' \xrightarrow{q_h, q_{h'}} Q'''_{h,h'}$ with $Q'_{h,h'} \equiv (\nu q'_1) \dots (\nu q'_m) (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) Q'''_{h,h'}$ (see proof of Proposition 4.2). Hence $\sigma =$
 $\{(\lambda_{1,h'}, [(\nu q'_1) \dots (\nu q'_m) (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) Q'''_{1,h'}]) \mid h' \in H_1\} \cup \dots \cup$
 $\{(\lambda_{n,h'}, [(\nu q'_1) \dots (\nu q'_m) (\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) Q'''_{n,h'}]) \mid h' \in H_n\}.$

Thus, considering the set I of pairs (h, h') , with $1 \leq h \leq n$ and $h' \in H_h$,

such that $[(\nu q'_1) \dots (\nu q'_m)(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) Q'''_{h,h'}]_{\equiv} = [P']_{\equiv}$ we have $\sum_{(\lambda, [P']_{\equiv}) \in \sigma} \lambda \cdot \mu_{\sigma}(\lambda, [P']_{\equiv}) = \sum_{(h,h') \in I} \lambda_{h,h'}$. Moreover I is also the set of pairs (h, h') such that $Q''_{h,h'} \equiv P'$, because $Q''_{h,h'} \equiv (\nu q'_1) \dots (\nu q'_m)(\nu q_1 \rightarrow \lambda_1) \dots (\nu q_n \rightarrow \lambda_n) Q'''_{h,h'}$. Therefore, since $(h, h') \neq (h'', h''')$ implies $id_{h,h'} \neq id_{h'',h'''}$, we also have $\sum_P \tau_{\lambda, id} \equiv P' \lambda = \sum_Q \tau_{\lambda, id} \equiv P' \lambda = \sum_{(h,h') \in I} \lambda_{h,h'}$. \square