

Structural Operational Semantics for Continuous State Probabilistic Processes

Giorgio Bacci, Marino Miculan

► **To cite this version:**

Giorgio Bacci, Marino Miculan. Structural Operational Semantics for Continuous State Probabilistic Processes. Dirk Pattinson; Lutz Schröder. 11th International Workshop on Coalgebraic Methods in Computer Science (CMCS), Mar 2012, Tallinn, Estonia. Springer, Lecture Notes in Computer Science, LNCS-7399, pp.71-89, 2012, Coalgebraic Methods in Computer Science. <10.1007/978-3-642-32784-1_5>. <hal-01539880>

HAL Id: hal-01539880

<https://hal.inria.fr/hal-01539880>

Submitted on 15 Jun 2017

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.



Structural operational semantics for continuous state probabilistic processes^{*}

Giorgio Bacci Marino Miculan

Dept. of Mathematics and Computer Science, University of Udine, Italy.
`giorgio.bacci@uniud.it`, `marino.miculan@uniud.it`

Abstract. We consider the problem of modeling syntax and semantics of probabilistic processes with continuous states (e.g. with continuous data). Syntax and semantics of these systems can be defined as algebras and coalgebras of suitable endofunctors over **Meas**, the category of measurable spaces. In order to give a more concrete representation for these coalgebras, we present an SOS-like rule format which induces an abstract GSOS over **Meas**; this format is proved to yield a fully abstract universal semantics, for which behavioural equivalence is a congruence. To this end, we solve several problems. In particular, the format has to specify how to compose the semantics of processes (which basically are continuous state Markov processes). This is achieved by defining a language of *measure terms*, i.e., expressions specifically designed for describing probabilistic measures. Thus, the transition relation associates processes with measure terms.

As an example application, we model a CCS-like calculus of processes placed in an Euclidean space. The approach we follow in this case can be readily adapted to other quantitative aspects, e.g. Quality of Service, physical and chemical parameters in biological systems, etc.

1 Introduction

Process algebras are widely used for compositional modeling of nondeterministic, communicating, mobile systems. Categorically, the syntax of processes is represented as the initial algebra of a signature functor, and their semantics as coalgebras of a suitable “behavioral” functor. According to the Structural Operational Semantics (SOS) paradigm [25], these coalgebras are described by means of labelled transition systems (LTSs), defined by induction on the syntactic structure of processes. In order to guarantee important properties about the resulting semantics, several formats of these SOS specifications have been studied. A well-known format is the so-called GSOS [8], which guarantees the bisimilarity to be a congruence (in most situations). Such a framework makes languages easier to understand, compare, and extend. In particular, a process algebra can be easily extended with new operators, without the need of time-consuming and error-prone proofs of congruence results.

^{*} Work supported by MIUR PRIN project 20088HXMYN, “SisteR”.

In recent years, this approach has been applied also to *stochastic* and *probabilistic* systems, due to their important applications to performance evaluation, systems biology, etc [19,9,18,15]. Bartels [6] and Klin and Sassone [20] have investigated rule formats (called Probabilistic GSOS and Stochastic GSOS, respectively) which guarantee bisimilarity to be a congruence.

However, these formats still do not cover the case of *continuous-state* (probabilistic) systems, like calculi with spatial/geometric features introduced in last years [10,4]. In these models, the behaviour of the system may be influenced by continuous data, which therefore is part of the state of the system. Typical examples are spatial informations, such as the position of processes and where transitions take place; e.g., in wireless networks distance may affect data access, or in biological models diffusion alters the signaling pathways, etc.

As a running example, in this paper we introduce a simple yet paradigmatic calculus of agents living in the Euclidean plane \mathbb{R}^2 , which we call *FlatCCS*¹. The idea we aim to model is that the probability of communications between two agents depends on their distance (like, e.g., in wireless networks). To this end, FlatCCS extends CCS (without restriction) with a syntactic “frame” operator representing a process’ displacement:

$$p, q ::= \text{nil} \mid \alpha.p \mid p + q \mid p \parallel q \mid [p]_z \qquad \alpha ::= a \mid \bar{a} \mid \tau$$

where a ranges over *actions*, and z over the plane \mathbb{R}^2 . Intuitively, if p is in position $z' \in \mathbb{R}^2$, the process $[p]_z$ is in $z' + z$. If no frame operator occurs, processes are assumed to be in the origin $(0,0)$. Thus, in $[p \parallel [q]_{(0,1)}]_{(1,0)}$, p is (externally) seen to be in $(1,0)$ and q in $(1,1)$.

As for the semantics for this calculus, here we assume that the communication probability decreases exponentially with the distance. Thus, we expect the process $a.\text{nil} \parallel [\bar{a}.\text{nil}]_{(r,0)}$ (with $r \in \mathbb{R}$) to perform a τ (that is, an internal communication) evolving into $\text{nil} \parallel [\text{nil}]_{(r,0)}$ with probability $e^{-|r|}$. The problem is how to specify this semantics.

In [6], labelled probabilistic transition systems are shown to be coalgebras $g: X \rightarrow (\mathcal{D}_\omega(X) + 1)^L$, where L is the set of labels, and \mathcal{D}_ω is the probability distribution functor over **Set** (which gives the set of *discrete* finite supported probability distributions over a given set). This behaviour functor leads to transitions of the form $p \xrightarrow{\alpha[r]} q$. However, we cannot use this approach for calculi like FlatCCS, because processes form a continuous space. This means that the probability of reaching any state q from p may be zero, yet the probability of reaching a subset of states may be nonzero.

The notion of interest is no longer a (discrete) probability distribution, but a (continuous) probability measure. Categorically, this corresponds to move to the category **Meas** of *measurable spaces* and *measurable functions*, and to model the a system behaviour by a coalgebra $g: X \rightarrow \Delta(X)^L$, where Δ is the Giry functor associating to X the set of probability measures over X , as advocated in [14,16,24]. This leads to transitions of the form $p \xrightarrow{\alpha} \mu$, where μ is a *measure* of the probabilistic distribution of the possible outcomes of p .

¹ Of course other variants can be considered, e.g. LineCCS, SpaceCCS, etc. [1].

Semantics with a similar transition format have been considered already in [11,3] for dealing with specific equational stochastic systems; these papers consider also suitable behavioural equivalences which are proved to be congruences. However, differently from the case of discrete (probabilistic) processes, SOS specifications and results in [11,3] are rather ad hoc, not based on any general framework for operational descriptions.

In this paper we aim to cover this gap, introducing a new GSOS rule format for probabilistic systems over measurable spaces. We will prove that this format guarantees that the resulting probabilistic behavioural equivalence is a congruence. As an example application, we will provide the semantics of FlatCCS according to this format.

To this end, we plan to apply the bialgebraic framework introduced by Turi and Plotkin [29]. However, in order to port this approach to our setting we have to solve several technical issues, due to the fact that we are working in **Meas** and using the Giry functor Δ . First, Δ does not preserve weak pullbacks [23,30], hence we cannot prove that bisimilarity is transitive and that it coincides with behavioural equivalence. As a consequence, we focus on behavioural equivalence instead of bisimilarity.

Secondly, **Meas** is not known to be Cartesian closed; hence, most of the constructions which can be carried out on **Set** and other toposes cannot be ported easily to **Meas**. In particular, we cannot follow Bartels’s approach for deriving a rule format from a distributive law [6].

Moreover, a “good” SOS rule format must be compositional, i.e., it has to define a system’s behaviour in terms of those of its subsystems. In traditional GSOS format, this is reflected by the fact that the target of a transition is a process built from the components of the source process, and their corresponding semantics. In our settings, the target of a transition is not a process term, but a measure over a generic measurable space, which do not have any syntactic structure to play with. In order to circumvent this problem, we propose to use transitions of the form $p \xrightarrow{\alpha} \mu$ where μ is a *measure term*, that is, a syntactic expression intended to denote a measure. The syntax of these measure terms, and their interpretation as measures, is part of the operational specification: a specification is given by a set of rules together with a description of how measures must be combined. We will show that this specification format, which we call *Measure GSOS specification format*, is general enough to cover the motivating example (and others [11,3]). In particular, we show that any LTS specification in this format leads to a distributive law of type $S(Id \times \Delta^L) \Rightarrow (\Delta T_S)^L$, where S is the syntactic functor and T_S the corresponding free monad. As a consequence, the induced behavioural equivalence is always a congruence.

Synopsis. In Section 2, we recall the coalgebraic presentation of continuous probabilistic systems in the category **Meas** using the Giry functor.

In Section 3 we describe how to define syntactic monads over **Meas**. A technical issue here is that polynomial functors (as those arising from syntactic signatures) are not known to preserve ω -colimits in **Meas**. To circumvent this

problem, we use a more general initial algebra construction, which requires mild conditions about the base category.

Then, in Section 4 we introduce the MGSOS specification format. Using the previous results, we show that an MGSOS specification corresponds to a distributive law of type $S(Id \times \Delta^L) \Rightarrow (\Delta T_S)^L$, and the induced behavioural equivalence is always a congruence. As an exemplification, all results of this section are applied to FlatCCS.

As mentioned above, a key feature of this format is the fact that target measures are described by means of a specific term language, which have to be interpreted as measures. In Section 5 we show how to construct these interpretation functions using a generalized induction proof principle.

Final remarks and conclusions are in Section 6.

Measure theoretic preliminaries. A σ -algebra over a set X is a non-empty family Σ_X of subsets of X closed under complements and countable unions. The pair (X, Σ_X) is called *measurable space* and the members of Σ_X are its *measurable sets*. A *family of generators* \mathcal{F} for Σ_X is a family of subsets of X such that the smallest σ -algebra containing \mathcal{F} is Σ_X , denoted by $\sigma(\mathcal{F}) = \Sigma_X$.

Let $(X, \Sigma_X), (Y, \Sigma_Y)$ be measurable spaces, a function $f: X \rightarrow Y$ is called *measurable* if $f^{-1}(E) = \{x \mid f(x) \in E\} \in \Sigma_X$, for all $E \in \Sigma_Y$ (notably, if Σ_Y is generated by \mathcal{F} , it suffices to show that $f^{-1}(F) \in \Sigma_X$, for all $F \in \mathcal{F}$ only).

A *(sub-)probability measure* on (X, Σ_X) is a function $\mu: \Sigma_X \rightarrow [0, 1]$ such that $\mu(X) = 1$ (resp. ≤ 1), and it is σ -additive, i.e. $\mu(\bigcup_{i \in I} E_i) = \sum_{i \in I} \mu(E_i)$ for all countable collections $\{E_i\}_{i \in I}$ of pairwise disjoint measurable sets in Σ_X .

The set of sub-probability measures over (X, Σ_X) forms a measurable space, denoted by ΔX , with σ -algebra generated by $\{\mathcal{B}_p(E) \mid p \in [0, 1], E \in \Sigma_X\}$, where $\mathcal{B}_p(E) = \{\mu \mid \mu(E) \geq p\}$.

2 Continuous probabilistic systems as coalgebras

The coalgebraic treatment of continuous probabilistic systems originates in the work of de Vink and Rutten [13]. Probabilistic bisimulation of Larsen and Skou [22] for discrete systems have been compared to the coalgebraic notion of bisimulation and ported without too much efforts to the continuous setting. Then, continuous probabilistic systems have been investigated from a coalgebraic point of view by Desharnais, Panagaden, Danos et al. [14,24,12], which provided the most of the work available on (labelled) Markov processes, together with Doberkat [16] on stochastic relations.

In this section we briefly recall labelled continuous probabilistic systems, i.e. generalized labelled Markov processes over measurable spaces of states, and some peculiarities about the category **Meas** (details can be found in [27]).

Definition 1. For a set L of action labels, a L -labelled Markov process on a measurable space (X, Σ_X) is a structure $(X, \{\tau_\alpha: X \times \Sigma_X \rightarrow [0, 1]\}_{\alpha \in L})$, where X is the set of states and, for each $\alpha \in L$, τ_α is a transition sub-probability

function, that is, a function such that, for all $x \in X$, $\tau_\alpha(x, \cdot)$ is a sub-probability measure, and, for each fixed $E \in \Sigma_X$, $\tau_\alpha(\cdot, E)$ is a measurable function.

Intuitively, $\tau_\alpha(x, E)$ is interpreted as the probability of the system starting in state x making a α -transition into one of the states in E . The transition probability is a conditional probability: it gives the probability of the system being in one of the states of the set E after the transition, provided that it was in the state x before the transition.

In order to model Markov processes as coalgebras one needs a suitable category and a suitable behaviour functor. The most natural choice for a category is **Meas**, the category of measurable spaces and measurable functions, and as for the behaviour functor, the Giry functor $\Delta: \mathbf{Meas} \rightarrow \mathbf{Meas}$, acting on objects as $X \mapsto \Delta X$ and on morphisms $f: X \rightarrow Y$ by $(\Delta f)(\mu) = \mu \circ f^{-1}$. This functor, first introduced by Lawvere, takes part in a monad triple commonly known as the Giry monad [17], with unit given by the Dirac measure δ_X , given by $\delta_X(x)(E) = 1$ if $x \in E$, $\delta_X(x)(E) = 0$ otherwise².

Proposition 2. *L-labelled Markov processes are exactly the Δ^L -coalgebras.*

For different reasons in most of the works on Markov processes different categories were considered. In [13], de Vink and Rutten used ultrametric spaces arguing that the main reason for doing so was reusing a theorem that guarantees existence of a final coalgebra for locally contractive functors. Another reason for avoiding **Meas** is that the Giry functor does not preserve weak-pullbacks [30], which would be desirable for bisimilarity to be well-behaved. For instance, Desharnais, Edelat et al. [14] moved to the categories of *analytic spaces* in order to construct semi-pullback, providing a way to show that bisimilarity is transitive.

In this work we remain in **Meas** and argue that for the coalgebraic treatment of Markov processes it suffices to work with general measurable spaces unless one prefers bisimilarity to behavioral equivalence (i.e., the relation given by pullbacks on final coalgebra homomorphisms). It is worthwhile to recall that in general, behavioural equivalence does not coincide with bisimilarity unless the behavior functor preserves weak pullbacks. This choice was already discussed by several authors, who have observed that in the generalized probabilistic setting, behavioral equivalence is more sensible than bisimilarity [12,7] and have suggested the use of co-congruence (also called event bisimulation) instead of the standard bisimulation. Similar arguments have been already discussed with more generality by Kurz in his doctoral thesis [21], and by Staton [28] where four different notions of bisimulation were investigated.

We conclude recalling some results that will be used later in the paper.

The category **Meas** is complete and cocomplete: limits and colimits are obtained as in **Set** and endowed, respectively, with initial and final σ -algebra w.r.t. their cone and cocone maps (indeed, the forgetful functor $U: \mathbf{Meas} \rightarrow \mathbf{Set}$ preserves both limits and colimits).

² In [17], Giry considers actual probability measures, but it will be convenient to work with sub-probabilities instead. All the results go through also in this extended case.

Completeness and cocompleteness of **Meas** allows us to consider the class of *polynomial endofunctors*, that is, the smallest class of endofunctors containing the identity Id , the constant functor \underline{M} for all measurable spaces M , and closed under binary product and coproduct.

Moss and Viglizzo [23,30] showed that polynomial functors extended with the Giry functor have a final coalgebra. Therefore, for L a finite set of action labels, we are allowed to safely adopt behavioural equivalence over Δ^L -coalgebras, since $\Delta^L \cong \prod_{\alpha \in L} \Delta$ is a finite product, hence admits a final coalgebra.

3 Syntactic monads over measurable spaces

A crucial component of the bialgebraic approach is the syntactical monad associated with the endofunctor induced by the term syntax. Given a functor $F: \mathcal{C} \rightarrow \mathcal{C}$, the free monad associated to F is defined on an object X by means of the initial algebra of the functor $Y \mapsto X + FY$. However, the usual construction of initial algebras as colimits of (initial) ω -sequences, due to Smyth and Plotkin [26], cannot be applied to polynomial endofunctors in **Meas**, because they are not known to preserve colimits of ω -sequences. However, we can adopt a generalization of [26] due to Adámek et al. [2] (see also Barr [5]), which extends to arbitrary ordinals the definition of *initial sequence* of F , where \mathcal{C} is assumed to have an initial object 0 and colimits along ordinal indexed diagrams.

The construction is well known but we recall it for sake of clarity. The initial sequence of F is an ordinal-indexed sequence of objects $(A_\beta)_{\beta \in \mathbf{Ord}}$ with arrows $(f_\beta^\gamma: A_\gamma \rightarrow A_\beta)_{\gamma \leq \beta}$, uniquely defined by the following conditions, for $\delta \leq \gamma \leq \beta$:

- IS-1. $A_{\beta+1} = FA_\beta$;
- IS-2. $f_{\beta+1}^{\gamma+1} = Ff_\beta^\gamma$;
- IS-3. $f_\beta^\beta = id_{A_\beta}$;
- IS-4. $f_\beta^\gamma \circ f_\gamma^\delta = f_\beta^\delta$;
- IS-5. if β is a limit ordinal, the cocone $(f_\beta^\gamma: A_\gamma \rightarrow A_\beta)_{\gamma < \beta}$ is a colimit.

The sequence is given by transfinite induction on $\alpha \in \mathbf{Ord}$, defining A_α and $f_\alpha^\beta: A_\beta \rightarrow A_\alpha$, for all $\beta \leq \alpha$, and checking at each stage that conditions (IS-1) to (IS-5) hold for the portion of sequence already defined.

First step: Let $\alpha = 0$. The sequence begins with $A_0 = 0$ and $f_0^0 = id_{A_0}$.

Isolated step: Let $\alpha = \alpha' + 1$ and assume by inductive hypothesis that $A_{\alpha'}$ and the arrows $f_{\alpha'}^\gamma$ have been given and satisfy (IS-1)–(IS-5), for all $\gamma < \alpha'$. We define $A_\alpha = FA_{\alpha'}$, and the arrows f_α^β are defined by induction on $\beta \leq \alpha$. We distinguish three cases. If $\beta = \alpha$, then we define $f_\alpha^\beta = id_{A_\alpha}$. If β is a successor ordinal, say $\beta = \beta' + 1$, then we define $f_\alpha^\beta = Ff_{\alpha'}^{\beta'}$. If β is a limit ordinal, $(f_\beta^\gamma: A_\gamma \rightarrow A_\beta)_{\gamma < \beta}$ is a colimit, by (IS-5). By inductive hypothesis on β , we can consider $(f_\alpha^\gamma: A_\gamma \rightarrow A_\alpha)_{\gamma < \beta}$, which turns out to be a compatible cocone by (IS-2) and (IS-4). Now, we define f_α^β as the unique map factorizing the cocone.

Limit step: Let α be a limit ordinal. By inductive hypothesis we are given all arrows f_β^γ , for $\gamma \leq \beta < \alpha$, which, indeed, form a chain. We define A_α to be the colimit of this chain and $(f_\alpha^\beta: A_\beta \rightarrow A_\alpha)_{\beta < \alpha}$ are its injections.

Although not explicitly shown, in each step of the above construction conditions (IS-1)–(IS-5) hold (see [5] for a detailed proof).

This sequence is defined through all ordinals, thus its construction does not terminate. If it should ever happen that $f_{\alpha+1}^\alpha$ is an isomorphism, it is clear that f_β^α is an isomorphism, for all $\beta > \alpha$. In this case we say that the sequence *stabilizes* (or *terminates*) at α . Note that, since $f_{\alpha+1}^\alpha$ is an isomorphism it has inverse $(f_{\alpha+1}^\alpha)^{-1}: FA_\alpha \rightarrow A_\alpha$, which defines the structure map of an F -algebra with carrier A_α . For this particular algebra we have the following result.

Theorem 3 ([5]). *Let \mathcal{C} be a category with initial object and colimits along ordinal-indexed diagrams, and $F: \mathcal{C} \rightarrow \mathcal{C}$ be a functor. Suppose the initial sequence of F stabilizes at α , then $(A_\alpha, (f_{\alpha+1}^\alpha)^{-1})$ is an initial F -algebra.*

In many situations it is difficult to determine the ordinal α at which the initial sequence stabilizes. However, if the category \mathcal{C} admits a factorization system, say $(\mathcal{E}, \mathcal{M})$, and it is \mathcal{E} -cowell-powered, the following result can be used instead.

Theorem 4. *Let $(\mathcal{E}, \mathcal{M})$ be a factorization system for a \mathcal{E} -cowell-powered category \mathcal{C} , and $F: \mathcal{C} \rightarrow \mathcal{C}$ be a functor that preserves \mathcal{E} -morphisms. Suppose the initial sequence of F has arrow $f_{\alpha+1}^\alpha: A_\alpha \rightarrow A_{\alpha+1}$ in \mathcal{E} , for some ordinal α , then an initial F -algebra exists.*

Proof. An easy transfinite induction argument shows that $f_\delta^\beta: A_\beta \rightarrow A_\delta$ are in \mathcal{E} , for all $\alpha \leq \beta \leq \delta$. The base case is clear. The successor step holds since F preserves \mathcal{E} -morphisms, and for the limit step, one uses that \mathcal{E} is closed by transfinite compositions. Hence we have that f_β^α are all \mathcal{E} quotients of A_α , for all $\beta \geq \alpha$. The result follows by the fact that \mathcal{C} is \mathcal{E} -cowell-powered, so the initial sequence of F stabilizes for some $\alpha' \leq \alpha$. \square

Remark 5. On **Meas**, there are many factorization systems that “lift” the factorization system on **Set**, e.g., quotients and injections form a factorization system, and so do surjections and subspace embeddings. In order to apply Theorem 4, however, we cannot choose the former factorization system. Indeed, polynomial functors in **Meas** do not preserve quotients (it is well known that binary products fail to do so), but they do preserve surjections.

Recall that polynomial functors in **Set** preserves colimits of ω -sequences, hence the initial sequence of these functors stabilizes at ω . Now, since the forgetful functor $U: \mathbf{Meas} \rightarrow \mathbf{Set}$ “maps” polynomial functors P in **Meas** to polynomial functors in **Set**, and reflects surjections, the arrow $f_{\omega+1}^\omega: A_\omega \rightarrow A_{\omega+1}$ in the initial sequence of P is surjective. Therefore, by Theorem 4, there exists an initial P -coalgebra.

Notably, the existence of initial P -algebras ensures that the forgetful functor $U^P: P\text{-Alg} \rightarrow \mathbf{Meas}$ has left adjoint $F^P \dashv U^P$. Therefore, one can construct the *freely generated monad* associated with P , denoted by (T_P, η^P, μ^P) . In the case P corresponds to a syntactic signature, this monad formally defines the measurable spaces of terms built over P , with *unit* $\eta^P: Id \Rightarrow T_P$ defined as the insertion-of-variables function, and *multiplication* $\mu^P: T_P^2 \Rightarrow T_P$ as the application-into-contexts function. Measurable sets for this space are induced by the cocone injections to the colimit (formally, to its sub-object) of the P -initial sequence.

4 Measure GSOS specification format

In this section we present a (very general) technique for constructing GSOS distributive laws of type $S(Id \times \Delta^L) \Rightarrow (\Delta T_S)^L$ via sets of inference rules for a fixed syntactic format. This gives rise to a universal semantics (both initial and final) which is *fully abstract* w.r.t. Δ^L -behavioural equivalence [29]. Moreover, the proposed method solves the problem of defining probability measures over continuous state spaces syntactically. This is done by using as targets of the transition relation, terms built over a new syntax, specifically designed for describing sub-probability measures.

Let \mathcal{S} and \mathcal{M} be two signatures, respectively called the *process* and the *measure terms signatures*. Let $S, M: \mathbf{Meas} \rightarrow \mathbf{Meas}$ be the two polynomial functors associated with those signatures, formally given by³

$$S = \coprod_{f \in \mathcal{S}} Id^{ar(f)} \quad M = \coprod_{f \in \mathcal{M}} Id^{ar(f)}.$$

By Remark 5, both S and M have their associated *freely generated monads*, namely, (T_S, η^s, μ^s) and (T_M, η^m, μ^m) . These monads define the measurable spaces of terms built over the signatures S and M , respectively.

Definition 6. *A MGSOS rule over the signatures S (for processes) and M (for measure terms), for a finite set L of labels, is an expression of the form*

$$\frac{\left\{ x_i \xrightarrow{a_{ij}} \mu_{ij} \right\}_{\substack{1 \leq j \leq m_i \\ 1 \leq i \leq n, a_{ij} \in A_i}} \quad \left\{ x_i \xrightarrow{b_i} \right\}_{\substack{b_i \in B_i \\ 1 \leq i \leq n}}}{f(x_1, \dots, x_n) \xrightarrow{c} \mu}$$

where

- $f \in S$ with $ar(f) = n$;
- $\{x_1, \dots, x_n\}$ and $\{\mu_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m_i\}$ are pairwise distinct variables;
- $A_i \cap B_i = \emptyset$ are disjoint subsets of labels in L , for all $1 \leq i \leq n$, and $c \in L$;
- μ is a measure term with variables in $\{x_1, \dots, x_n\}$ and $\{\mu_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m_i\}$.

Note that, differently from the standard GSOS rule format of [8], in the premises one is not allowed to use the same label twice for the same variable x_i ; moreover, the source and target occurring in the conclusion have different signatures.

Together with each set of MGSOS rules we assume a natural transformation of type $T_M \Delta \Rightarrow \Delta T_S$, which interprets measure terms occurring in the target of the transitions as actual measures over the space of processes. This allows for a richer expressivity in the definition of a structural operational semantics.

Definition 7. *A MGSOS specification system is a pair $(\mathcal{R}, \langle \cdot \rangle)$, such that \mathcal{R} is a set of image finite MGSOS rules, and $\langle \cdot \rangle: T_M \Delta \Rightarrow \Delta T_S$ is a natural transformation, called measure terms evaluation.*

³ By an abuse of notation, in the rest of the paper we will often use the associated functor in place of its signature.

Example 8. Let define the probabilistic semantics for FlatCCS by means of an MGSOS specification system $(\mathcal{R}^{\text{fl}}, \langle \cdot \rangle^{\text{fl}})$. The syntactic functors for FlatCCS processes and measure terms are given as follows, for a finite set A of labels:

$$\begin{aligned} SX &= \overbrace{1}^{\text{nil}} + \overbrace{A \times X}^{a.x} + \overbrace{A \times X}^{\bar{a}.x} + \overbrace{X}^{\tau.x} + \overbrace{X \times X}^{x+x} + \overbrace{X \times X}^{x\|x} + \overbrace{\mathbb{R}^2 \times X}^{[x]_z}, \\ MX &= \overbrace{X \times X}^{x \blacktriangleleft x} + \overbrace{X \times X}^{x \blacktriangleright x} + \overbrace{X \times X}^{x \blacktriangledown x} + \overbrace{\mathbb{R}^2 \times X}^{\langle x \rangle_z}. \end{aligned}$$

The set \mathcal{R}^{fl} of MGSOS rules gives the specification for a collective operational semantics in the form of a labelled transition system. A transition $p \xrightarrow{\alpha} \mu$ has a process p as source and a *measure term* μ as target, for $\alpha \in A \cup \bar{A} \cup \{\tau\}$.

$$\begin{array}{c} \frac{}{\alpha.x \xrightarrow{\alpha} x} \quad \frac{x \xrightarrow{\alpha} \mu}{x + x' \xrightarrow{\alpha} \mu} \quad \frac{x' \xrightarrow{\alpha} \mu}{x + x' \xrightarrow{\alpha} \mu} \\ \frac{x \xrightarrow{\alpha} \mu}{[x]_z \xrightarrow{\alpha} \langle \mu \rangle_z} \quad \frac{x \xrightarrow{\alpha} \mu}{x \parallel x' \xrightarrow{\alpha} \mu \blacktriangleleft x'} \quad \frac{x' \xrightarrow{\alpha} \mu}{x \parallel x' \xrightarrow{\alpha} x \blacktriangleright \mu} \\ \frac{x \xrightarrow{a} \mu \quad x' \xrightarrow{\bar{a}} \mu'}{x \parallel x' \xrightarrow{\tau} \mu \blacktriangledown \mu'} \quad \frac{x \xrightarrow{\bar{a}} \mu \quad x' \xrightarrow{a} \mu'}{x \parallel x' \xrightarrow{\tau} \mu \blacktriangledown \mu'} \end{array}$$

According to the rules above, $\mu \blacktriangleleft \mu'$ indicates that an action has been performed on the left hand side (dually in $\mu \blacktriangleright \mu'$), $\mu \blacktriangledown \mu'$ denotes that the process succeeded in a synchronization, and $\langle \mu \rangle_z$ encodes the absolute position.

The semantics intuitively given in Section 1 is formalized by means of the measure term evaluation $\langle \cdot \rangle^{\text{fl}}: T_M \Delta \Rightarrow \Delta T_S$, defined, for $\mu, \mu' \in T_M \Delta X$ and $\beta \in \Delta X$, as follows:⁴

$$\begin{aligned} \langle \beta \rangle_X^{\text{fl}} &= \beta \\ \langle \mu \blacktriangleleft \mu' \rangle_X^{\text{fl}} &= (\langle \mu \rangle_X^{\text{fl}} \times \langle \mu' \rangle_X^{\text{fl}}) \circ (\lambda(x, x'). x \parallel x')^{-1} \\ \langle \mu \blacktriangleright \mu' \rangle_X^{\text{fl}} &= (\langle \mu \rangle_X^{\text{fl}} \times \langle \mu' \rangle_X^{\text{fl}}) \circ (\lambda(x, x'). x \parallel x')^{-1} \\ \langle \mu \blacktriangledown \mu' \rangle_X^{\text{fl}} &= \left(e^{-\|pos(\mu) - pos(\mu')\|} \cdot (\langle \mu \rangle_X^{\text{fl}} \times \langle \mu' \rangle_X^{\text{fl}}) \right) \circ (\lambda(x, x'). x \parallel x')^{-1} \\ \langle \langle \mu \rangle_z \rangle_X^{\text{fl}} &= \langle \mu \rangle_X^{\text{fl}} \circ (\lambda x. [x]_z)^{-1}, \end{aligned}$$

where $pos: T_M \Delta X \rightarrow \mathbb{R}^2$ determines the position of an action by inspecting the syntactic structure of μ , inductively defined by

$$\begin{aligned} pos(\beta) &= (0, 0) & pos(\mu \blacktriangleleft \mu') &= pos(\mu) & pos(\mu \blacktriangleright \mu') &= pos(\mu') \\ pos(\mu \blacktriangledown \mu') &= \frac{1}{2}(pos(\mu) + pos(\mu')) & pos(\langle \mu \rangle_z) &= z + pos(\mu). \end{aligned}$$

Note that the use of a new syntax in the target of a transition is essential in order to recover the absolute position of the action that has been performed. ■

⁴ Here, $\beta \times \beta'$ denotes the measure product, and $(r \cdot \beta)(E) := r\beta(E)$, for $0 \leq r \leq 1$.

Similarly to GSOS transition systems specifications, also MGSOS specification systems define a structural operational semantics, but in this particular case in the form of a labelled Markov process over the measurable space of process terms. Its definition can be summarized in two stages. First, an image finite labelled transition system $(T_S0, \{\overset{\alpha}{\rightarrow} \subseteq T_S0 \times T_M(T_S0)\}_{\alpha \in L})$ is defined using the set of MGSOS derivation rules, then the associated Markov process is obtained evaluating measure terms to sub-probability measures. Formally, the associated Δ^L -coalgebra γ over T_S0 is defined, for $\alpha \in L$ and $p \in T_S0$, by

$$\gamma(p)(\alpha) = \oplus_{T_S0}(\{\langle \mu \rangle_{T_S0} \mid p \overset{\alpha}{\rightarrow} \mu\}).$$

where, for a finite set of U sub-probability measures over X , we define $\oplus_X(U): \Sigma_X \rightarrow [0, 1]$ by $\oplus_X(U)(E) = \sum_{\mu \in U} \mu(E) / \sum_{\mu \in U} \mu(X)$, which is easily seen to be a (sub-)probability measure over X , and will be called *(weighted) sum of sub-probability measures* in U .

The main advantage in using this two staged definition is that Markov processes are defined syntactically by structural induction on process terms.

Example 9 (Trilateration). Let us see how the specification system for FlatCCS given in Example 8 may be used to model the *node positioning problem* using a trilateration strategy (as in GPS, robot position detection, etc). In order to determine its position, an agent calculates the distances of three (or more) beacon reference stations (anchors), whose positions are known. These distances can be derived from the communication probability between the agents and the anchors. A simple FlatCCS system implementing this scenario is the following:

$$\begin{aligned} Sys &= (Agent \parallel Anchs) \\ Agent &= [(\bar{a}_1 \parallel \bar{a}_2) \parallel \bar{a}_3]_{(x,y)}, \\ Anchs &= ([a_1]_{(x_1,y_1)} \parallel [a_2]_{(x_2,y_2)}) \parallel [a_3]_{(x_3,y_3)}. \end{aligned}$$

There are only three possible transitions of the form $Sys \xrightarrow{\tau} \mu_i$, namely:

$$\begin{aligned} \mu_1 &= \langle (\mathbf{nil} \blacktriangleleft \bar{a}_2) \blacktriangleleft \bar{a}_3 \rangle_{(x,y)} \blacktriangledown ((\langle \mathbf{nil} \rangle_{(x_1,y_1)} \blacktriangleleft \langle a_2 \rangle_{(x_2,y_2)}) \blacktriangleleft \langle a_3 \rangle_{(x_3,y_3)}), \\ \mu_2 &= \langle (\bar{a}_1 \blacktriangleright \mathbf{nil}) \blacktriangleleft \bar{a}_3 \rangle_{(x,y)} \blacktriangledown ((\langle a_1 \rangle_{(x_1,y_1)} \blacktriangleright \langle \mathbf{nil} \rangle_{(x_2,y_2)}) \blacktriangleleft \langle a_3 \rangle_{(x_3,y_3)}), \\ \mu_3 &= \langle (\bar{a}_1 \parallel \bar{a}_2) \blacktriangleright \mathbf{nil} \rangle_{(x,y)} \blacktriangledown (([a_1]_{(x_1,y_1)} \parallel [a_2]_{(x_2,y_2)}) \blacktriangleright \langle \mathbf{nil} \rangle_{(x_3,y_3)}), \end{aligned}$$

from which we can get the distances Δ_i from each anchor node $[a_i]_{(x_i,y_i)}$. In order to do so, for $\vec{E} = (E, E_1, E_2, E_3)$ a quadruple of measurable sets in \mathbb{R}^2 , define the following measurable sets in T_S0 ,

$$\begin{aligned} \Gamma_1(\vec{E}) &= \{[(\mathbf{nil} \parallel \bar{a}_2) \parallel \bar{a}_3]_e \mid (([\mathbf{nil}]_{e_1} \parallel [a_2]_{e_2}) \parallel [a_3]_{e_3}) \mid e \in E, e_i \in E_i\}, \\ \Gamma_2(\vec{E}) &= \{[(\bar{a}_1 \parallel \mathbf{nil}) \parallel \bar{a}_3]_e \mid (([a_1]_{e_1} \parallel [\mathbf{nil}]_{e_2}) \parallel [a_3]_{e_3}) \mid e \in E, e_i \in E_i\}, \\ \Gamma_3(\vec{E}) &= \{[(\bar{a}_1 \parallel \bar{a}_2) \parallel \mathbf{nil}]_e \mid (([a_1]_{e_1} \parallel [a_2]_{e_2}) \parallel [\mathbf{nil}]_{e_3}) \mid e \in E, e_i \in E_i\}. \end{aligned}$$

Exploiting the definition of $\langle \cdot \rangle_{T_S0}^{\text{fl}}$ in Example 8 we have $\langle \mu_i \rangle_{T_S0}^{\text{fl}}(\Gamma_i(\vec{E})) = e^{-\Delta_i}$ whenever $(x, y) \in E$ and $(x_j, y_j) \in E_j$, for $1 \leq j \leq 3$ (otherwise is 0). Now, the

coordinates x and y of the agent can be determined as the intersection of three circles with center in (x_i, y_i) and radius Δ_i , whose solution is given by⁵

$$y = \frac{2(V_{1,2}(x_2 - x_3) - V_{3,2}(x_2 - x_1))}{(y_1 - y_2)(x_2 - x_3) - (y_3 - y_2)(x_2 - x_3)}$$

$$x = \frac{y(y_1 - y_2) - 2V_{1,2}}{x_2 - x_3}$$

where $V_{i,j} = (x_j^2 - x_i^2) + (y_j^2 - y_i^2) + (\Delta_i^2 - \Delta_j^2)$, for $1 \leq i, j \leq 3$. Note that, in a more general situation, the position (x, y) can only be stochastically determined with precision depending on the chosen measurable sets to be tested. ■

The next theorem shows how MGSOS specification systems induce natural transformation of type $S(Id \times \Delta^L) \Rightarrow (\Delta T_S)^L$, i.e., abstract GSOS laws of [29].

Theorem 10. *An MGSOS specification system $(\mathcal{R}, \langle \cdot \rangle)$ over S and M , and set of labels L , determines a natural transformation of type $S(Id \times \Delta^L) \Rightarrow (\Delta T_S)^L$.*

Proof. For any space X , we define $\llbracket \mathcal{R} \rrbracket_X : S(X \times (\Delta X)^L) \rightarrow (\Delta T_S X)^L$ in two steps as the composite of two (set) functions:

$$S(X \times (\Delta X)^L) \xrightarrow{\nu_X} (\mathcal{P}_{\text{fin}} T_M \Delta X)^L \xrightarrow{(\oplus T_S \circ \mathcal{P}_{\text{fin}} \langle \cdot \rangle)_X^L} (\Delta T_S X)^L,$$

where, for every space X , ν_X is defined as follows: for all $t \in T_M \Delta X$, $f \in S$, $c \in L$, $x_i \in X$, and $\beta_i \in (\Delta X)^L$, put

$$t \in \nu_X(f((x_1, \beta_1), \dots, (x_n, \beta_n)))(c)$$

if and only if there exists a (possible renamed) rule in \mathcal{R} with conclusion of the form $f(x_1, \dots, x_n) \xrightarrow{c} \mu$, such that $\beta_i(b) = \omega$, for $b \in B_i$, and $t = (T_M \sigma)(\mu)$ for a substitution map σ such that $\sigma(x_i) = \delta_{x_i}$ and $\sigma(\mu_{ij}) = \beta_i(a_{ij})$ (the indices i, j follows the conventions of Definition 6).

Naturality of $\llbracket \mathcal{R} \rrbracket$ is proved separately for the two components. Since $\langle \cdot \rangle$ and \oplus are natural, the second component is easily seen to be natural as well; whereas for naturality of ν one proceeds as in [29, Th. 1.1].

As for measurability of $\llbracket \mathcal{R} \rrbracket_X$, it suffices to check that $\llbracket \mathcal{R} \rrbracket_X^{-1}(U_\alpha[E])$ is measurable in $S(X \times (\Delta X)^L)$, for $U_\alpha[E] = \{\beta' \in (\Delta T_S X)^L \mid \beta'(\alpha) \in E\}$, with $\alpha \in L$, and $E \in \Sigma_{\Delta T_S X}$.

⁵ In general, this is a systems of three independent non-linear equations, hence it may have multiple or no real solutions. Here, each radius is determined by measuring the distance from the intersecting point, therefore it has always a unique solution.

Let $f(\overline{x, \beta})$ be an abbreviation for $f((x_1, \beta_1), \dots, (x_n, \beta_n)) \in S(X \times (\Delta X)^L)$, then we have

$$\begin{aligned} \llbracket \mathcal{R} \rrbracket_X^{-1}(U_\alpha[E]) &= \left\{ f(\overline{x, \beta}) \mid \llbracket \mathcal{R} \rrbracket_X(f(\overline{x, \beta})) \in U_\alpha[E] \right\} \\ &= \left\{ f(\overline{x, \beta}) \mid ((\oplus T_S \circ \mathcal{P}_{\text{fin}} \langle \cdot \rangle)^L \circ \nu)_X(f(\overline{x, \beta})) \in U_\alpha[E] \right\} \\ &= \left\{ f(\overline{x, \beta}) \mid ((\oplus T_S \circ \mathcal{P}_{\text{fin}} \langle \cdot \rangle)^L \circ \nu)_X(f(\overline{x, \beta}))(\alpha) \in E \right\} \\ &= \left\{ f(\overline{x, \beta}) \mid (\oplus T_S \circ \mathcal{P}_{\text{fin}} \langle \cdot \rangle \circ \nu^\alpha)_X(f(\overline{x, \beta})) \in E \right\} \end{aligned}$$

where $\nu_X^\alpha \triangleq \nu_X(\cdot)(\alpha)$ is the specialization of ν_X on a fixed $\alpha \in L$,

$$\begin{aligned} &= (\oplus T_S \circ \mathcal{P}_{\text{fin}} \langle \cdot \rangle \circ \nu^\alpha)_X^{-1}(E) \\ &= (\nu_X^\alpha)^{-1} \circ (\oplus T_S \circ \mathcal{P}_{\text{fin}} \langle \cdot \rangle)_X^{-1}(E) \end{aligned}$$

Now is easy to prove measurability. Since $\langle \cdot \rangle$ is measurable and sums and products of measurable functions is measurable, $(\oplus T_S \circ \mathcal{P}_{\text{fin}} \langle \cdot \rangle)_X^{-1}(E) \in \Sigma_{\mathcal{P}_{\text{fin}} T_M \Delta X}$.

To prove measurability of ν_X^α we need only to check that $(\nu_X^\alpha)^{-1}(\bigcup_{j=0}^k \{E_j\})$ is a measurable set in $S(X \times (\Delta X)^L)$, for $E_j \in \Sigma_{T_M \Delta X}$, $0 \leq j \leq k$.

$$\begin{aligned} (\nu_X^\alpha)^{-1}(\bigcup_{j=0}^k \{E_j\}) &= \left\{ f(\overline{x, \beta}) \mid \nu_X^\alpha(f(\overline{x, \beta})) \in \bigcup_{j=0}^k \{E_j\} \right\} \\ &= \bigcup_{j=0}^k \left\{ f(\overline{x, \beta}) \mid \nu_X^\alpha(f(\overline{x, \beta})) = E_j \right\} \end{aligned}$$

But $\nu_X^\alpha(f(\overline{x, \beta})) = E_j$ iff there exists some rule in \mathcal{R} with conclusion of the form $f(x_1, \dots, x_n) \xrightarrow{\alpha} \mu$ and $(T_M \sigma)(\mu) \in E_j$ (obviously, all the other conditions given above have to be satisfied too). By construction, σ is defined by sums of Dirac measures δ_X and $ev_{a_{ij}}$, which are measurable. Therefore, $T_M \sigma$ is measurable, and as a consequence also ν_X^α is measurable. \square

In the above proof, measure terms variables are interpreted as Dirac measures via the natural transformation $\delta: Id \Rightarrow \Delta$ (hence, measurable in each component). This together with the assumption that $\langle \cdot \rangle: T_M \Delta \Rightarrow \Delta T_S$ is a natural transformation are crucial to prove measurability of $\llbracket \mathcal{R} \rrbracket$.

Example 11. For the MGSOS specification system given in Example 8 the construction in Theorem 10, for $x, x' \in X$ and $\beta, \beta' \in (\Delta X)^L$, gives

$$\begin{aligned} \llbracket \text{nil} \rrbracket_X &= \lambda \alpha. \omega \\ \llbracket \alpha. \rrbracket_X(x, \beta) &= \lambda \alpha'. \begin{cases} \langle \delta_x \rangle_X^{\text{fl}} & \text{if } \alpha' = \alpha \\ \omega & \text{otherwise} \end{cases} \\ (x, \beta) \llbracket + \rrbracket_X(x', \beta') &= \lambda \alpha. \langle \beta(\alpha) \rangle_X^{\text{fl}} \oplus \langle \beta'(\alpha) \rangle_X^{\text{fl}} \\ (x, \beta) \llbracket \parallel \rrbracket_X(x', \beta') &= \lambda \alpha. \begin{cases} \langle \beta(\alpha) \blacktriangleleft \delta_{x'} \rangle_X^{\text{fl}} \oplus \langle \delta_x \blacktriangleright \beta'(\alpha) \rangle_X^{\text{fl}} & \text{if } \alpha \neq \tau \\ \bigoplus_{a \in A} \left\{ \langle \beta(\alpha) \blacktriangleleft \delta_{x'} \rangle_X^{\text{fl}}, \langle \delta_x \blacktriangleright \beta'(\alpha) \rangle_X^{\text{fl}}, \right. \\ \left. \langle \beta(\bar{a}) \blacktriangleright \beta'(\bar{a}) \rangle_X^{\text{fl}}, \langle \beta(\bar{a}) \blacktriangleleft \delta_{x'} \rangle_X^{\text{fl}} \right\} & \text{if } \alpha = \tau \end{cases} \\ \llbracket []_z \rrbracket_X(x, \beta) &= \lambda \alpha'. \langle \beta(\alpha) \rangle_X^{\text{fl}} \end{aligned}$$

where ω is the null sub-probability measure assigning to each measurable set probability zero. Note that, since A is finite, the last summand is well defined.

When the measurable space of variables X is taken to be the space $T_S 0$ of ground measurable terms, the above definition gives the “canonical” initial denotational semantics for FlatCCS. ■

In [29], it is shown that distributive laws $\rho: \Sigma(Id \times B) \Rightarrow BT_\Sigma$, where T_Σ is the free monad over Σ , for endofunctors Σ and B admitting initial Σ -algebra $(T_\Sigma 0, \alpha)$ and final B -coalgebra (F, ω) , give rise to a unique B -coalgebra structure $\beta_\rho: T_\Sigma 0 \rightarrow BT_\Sigma 0$ such that $(T_\Sigma 0, \alpha, \beta_\rho)$ is the initial ρ -model (i.e., the initial ρ -bialgebra). Dually, there is a unique Σ -algebra structure $\alpha_\rho: \Sigma F \rightarrow F$ such that (F, α_ρ, β) is the final ρ -model. Thus, the following holds:

Theorem 12 (Adequacy [29]). *Let ρ be an abstract distributive GSOS law. The unique (both by initiality and finality) homomorphism from the initial to the final ρ -model is both the initial and final semantics for ρ .* □

Due to this result, the above is called *universal semantics* for ρ . Note, that two B -coalgebras have the same universal semantics if and only if they are behavioural equivalent, therefore B -behavioural equivalence is a Σ -congruence.

As for abstract GSOS laws of type $S(Id \times \Delta^L) \Rightarrow (\Delta T_S)^L$ we have the following result for behavioural equivalence on probabilistic processes on **Meas**.

Corollary 13. *Behavioural equivalence on the Δ^L -coalgebras over $T_S 0$ inductively induced by MGSOS specification systems over S and L , is a S -congruence.*

Proof. By Theorem 4 and Remark 5, every syntactic functor $S: \mathbf{Meas} \rightarrow \mathbf{Meas}$ have initial algebra. As for the endofunctor $\Delta^L: \mathbf{Meas} \rightarrow \mathbf{Meas}$, a proof of the existence of the final coalgebra is given in [30]. The thesis follows combining Theorems 10 and 12. □

5 Measure terms evaluations

MGSOS specifications require the evaluations $\langle \cdot \rangle_X: T_M \Delta X \rightarrow \Delta T_S X$ to be both natural and measurable. This time consuming check is overcome in this section, where we provide a rather easy (and general) technique for defining natural measure terms evaluation functions.

This technique is based on a generalized induction proof principle, dual to the “coiterative proof principle” described in [6]. This method is then shown to work for the definition of measure terms evaluation for FlatCCS of Example 8.

We start giving a recursion lemma dual to [6], which generalizes the standard induction proof principle by means of a simple distributive law λ .

Lemma 14. *Let $S, B: \mathcal{C} \rightarrow \mathcal{C}$ be functors on a category with countable products, (A, α) be the initial S -algebra, and $\lambda: SB \Rightarrow BS$ be a (simple) distributive law.*

For any SB -algebra (X, φ) there exists a unique λ -iterative arrow $f: A \rightarrow X$ induced by φ , such that the following diagrams commute

$$\begin{array}{ccc}
 SBX & \xleftarrow{SBf} & SBA \xleftarrow{S\beta_\lambda} SA \\
 \varphi \downarrow & & \downarrow \alpha \\
 X & \xleftarrow{f} & A
 \end{array}
 \qquad
 \begin{array}{ccccc}
 SA & \xrightarrow{\alpha} & A & \xrightarrow{\beta_\lambda} & BA \\
 S\beta_\lambda \downarrow & & & & \uparrow B\alpha \\
 SBA & \xrightarrow{\lambda_A} & & & BSA
 \end{array}$$

Proof. Dualize [6, Theorem 4.2.2].

We denote this induction proof principle by λ -iteration proof principle, and we call $f: A \rightarrow X$ as the λ -iterative arrow induced by φ . Note that, the diagram on the right is the initial λ -model, and in particular β_λ is uniquely determined by (standard) induction on the S -algebra $(BA, B\alpha \circ \lambda_A)$.

The λ -iteration proof principle of Lemma 14 can be extended as a proof principle on the monad T_S freely generated by S as follows:

Proposition 15 (Structural λ -iteration). *Let $S, B: \mathcal{C} \rightarrow \mathcal{C}$ be functors on a category with binary coproducts and countable products, (T_S, η^s, μ^s) be the free monad over S , $\lambda: SB \Rightarrow BS$ be a distributive law, and $\psi_X: ST_S X \rightarrow T_S X$ be the free S -algebra structure over X . For any SB -algebra (Y, φ) , B -coalgebra (X, k) , and arrow $\phi: X \rightarrow Y$, there exists a unique arrow $f: T_S X \rightarrow Y$ such that the following diagrams commute*

$$\begin{array}{ccc}
 SBY & \xleftarrow{SBf} & SBT_S X \xleftarrow{S\beta_\lambda} ST_S X \\
 \varphi \downarrow & & \downarrow \psi_X \\
 Y & \xleftarrow{f} & T_S X \\
 & \nearrow \phi & \uparrow \eta_X^s \\
 & & X
 \end{array}
 \qquad
 \begin{array}{ccccc}
 X & \xrightarrow{\eta_X^s} & T_S X & \xleftarrow{\psi_X} & ST_S X \\
 k \downarrow & & \beta_\lambda \downarrow & & \downarrow S\beta_\lambda \\
 BX & \xrightarrow{B\eta_X^s} & BT_S X & \xleftarrow{B\psi_X \circ \lambda_{T_S X}} & SBT_S X
 \end{array}$$

Proof. First, notice that β_λ is the inductive extension of the $(\underline{X} + S)$ -algebra structure on $BT_S X$ given by the copair $[B\eta_X^s \circ k, B\psi_X \circ \lambda_{T_S X}]$, along the initial $(\underline{X} + S)$ -algebra structure $[\eta_X^s, \psi_X]$ on $T_S X$. Now, define the distributive law $\lambda': (\underline{X} + S)B \Rightarrow B(\underline{X} + S)$ as $\lambda'_Y = [Bin_l, Bin_r] \circ (k + \lambda_Y)$ (the proof of naturality is straightforward). By definition of λ' we have

$$[B\eta_X^s \circ k, B\psi_X \circ \lambda_{T_S X}] = B[\eta_X^s, \psi_X] \circ \lambda'_{T_S X}.$$

Therefore, by unicity of the inductive extension, $(T_S X, \psi_X, \beta_\lambda)$ turns out to be a λ' -model on $T_S X$. This allows to apply Lemma 14 on the $(\underline{X} + S)B$ -algebra structure $(Y, [\phi, \varphi])$ obtaining a unique λ' -iterative arrow $f: T_S X \rightarrow Y$ making the required diagrams above commute. \square

We denote this proof principle by *structural λ -induction proof principle*, and we say that f is the λ -iterative extension of φ along the (pair of) valuation functions ϕ and k . Note that, the diagram on the right define β_λ as the structural inductive extension of $B\psi_X \circ \lambda_{T_S X}$ along the valuation function $B\eta_X^s \circ k$.

Proposition 15 can be turned into an induction proof principle on natural transformations in the following way:

Corollary 16. *Let $S, B, F: \mathcal{C} \rightarrow \mathcal{C}$ be functors on a category with binary coproducts and countable products, (T_S, η^s, μ^s) be the free monad over S , $\lambda: SB \Rightarrow BS$ be a distributive law, and $\psi: ST_S \Rightarrow T_S$ be the natural transformation defined on any component X as the free S -algebra structures over X . For any $\varphi: SBF \Rightarrow F$, $k: Id \Rightarrow B$, and $\phi: Id \Rightarrow F$, there exists a unique natural transformation $f: T_S \Rightarrow F$ such that the following (natural) diagrams commute*

$$\begin{array}{ccc}
 SBF & \xleftarrow{SBf} & SBT_S \xleftarrow{S\beta_\lambda} ST_S \\
 \varphi \downarrow & & \downarrow \psi \\
 F & \xleftarrow{f} & T_S \\
 & \nearrow \phi & \uparrow \eta^s \\
 & & Id
 \end{array}
 \qquad
 \begin{array}{ccccc}
 Id & \xrightarrow{\eta^s} & T_S & \xleftarrow{\psi} & ST_S \\
 k \downarrow & & \beta_\lambda \downarrow & & \downarrow S\beta_\lambda \\
 B & \xrightarrow{B\eta^s} & BT_S & \xleftarrow{B\psi \circ \lambda_{T_S}} & SBT_S
 \end{array}$$

where $\beta_\lambda: T_S \Rightarrow BT_S$ is defined on any component X as the unique inductive extension of $B\psi_X \circ \lambda_{T_S X}$ along the valuation function $B\eta_X^s \circ k_X$.

Proof. We first prove naturality of β_λ , i.e., that for any morphism $g: X \rightarrow Y$, $(\beta_\lambda)_Y \circ T_S g = BT_S g \circ (\beta_\lambda)_X$. The commuting diagrams

$$\begin{array}{ccccc}
 X & \xrightarrow{\eta_X^s} & T_S X & \xleftarrow{\psi_X} & ST_S X \\
 k_X \downarrow & \text{(by def.)} & (\beta_\lambda)_X \downarrow & \text{(by def.)} & \downarrow S(\beta_\lambda)_X \\
 BX & \xrightarrow{B\eta_X^s} & BT_S X & \xleftarrow{B\psi_X \circ \lambda_{T_S X}} & SBT_S X \\
 Bg \downarrow & \text{(nat. } \eta^s) & BT_S g \downarrow & \text{(nat. } \psi \text{ and } \lambda) & \downarrow SBT_S g \\
 BY & \xrightarrow{B\eta_Y^s} & BT_S Y & \xleftarrow{B\psi_Y \circ \lambda_{T_S Y}} & SBT_S Y
 \end{array}$$

and

$$\begin{array}{ccccc}
 & & X & \xrightarrow{\eta_X^s} & T_S X & \xleftarrow{\psi_X} & ST_S X \\
 & & \swarrow k_X & & \downarrow T_S g & & \downarrow ST_S g \\
 & & Y & \xrightarrow{\eta_Y^s} & T_S Y & \xleftarrow{\psi_Y} & ST_S Y \\
 & & \text{(nat. } k) & & \text{(nat. } \psi) & & \\
 & & \downarrow g & & \downarrow (\beta_\lambda)_Y & & \downarrow S(\beta_\lambda)_Y \\
 BX & & Y & \xrightarrow{\eta_Y^s} & T_S Y & \xleftarrow{\psi_Y} & ST_S Y \\
 & & \text{(nat. } k) & & \text{(by def.)} & & \text{(by def.)} \\
 & & \downarrow Bg & & \downarrow BT_S g & & \downarrow SBT_S g \\
 & & BY & \xrightarrow{B\eta_Y^s} & BT_S Y & \xleftarrow{B\psi_Y \circ \lambda_{T_S Y}} & SBT_S Y \\
 & & \text{(nat. } k) & & \text{(by def.)} & & \text{(by def.)}
 \end{array}$$

assert that $(\beta_\lambda)_Y \circ T_S g$ and $BT_S g \circ (\beta_\lambda)_X$ are both inductive extensions of $B\psi_Y \circ \lambda_{T_S Y}$ along $B\eta_X^s \circ Bg \circ k_X$, hence they necessarily coincide.

Naturality of f follows similarly by unicity of the λ -iterative extension. Indeed, both $f_Y \circ Tg$ and $Fg \circ f_X$ are λ -iterative extensions of φ_Y along the pair of valuations $Fg \circ \phi_X$ and k_X , as proved by the commutative diagrams

$$\begin{array}{ccccccc}
 & & & & SBT_S g & & SBT_S X \\
 & & & & \curvearrowright & & \uparrow S(\beta_\lambda)_X \\
 SBFY & \xleftarrow{SBf_Y} & SBT_S Y & \xleftarrow{S(\beta_\lambda)_Y} & ST_S Y & \xleftarrow{ST_S g} & ST_S X \\
 \varphi_Y \downarrow & & \text{(by def.)} & & \psi_Y \downarrow & \text{(nat. } \psi) & \psi_X \downarrow \\
 FY & \xleftarrow{f_Y} & T_S Y & \xleftarrow{T_S g} & T_S X & & \\
 Fg \uparrow & & \phi_Y & \text{(nat. } \phi) & \eta_Y^s \uparrow & \text{(nat. } \eta^s) & \eta_X^s \uparrow \\
 FX & \xleftarrow{\phi_X} & Y & \xleftarrow{g} & X & & \\
 & & & & & &
 \end{array}$$

and

$$\begin{array}{ccccccc}
 SBFY & \xleftarrow{SBf_g} & SBFX & \xleftarrow{SBf_X} & SBT_S X & \xleftarrow{S(\beta_\lambda)_X} & ST_S X \\
 \varphi_Y \downarrow & & \text{(nat. } \varphi) & \downarrow \varphi_X & \text{(by def.)} & & \psi_X \downarrow \\
 FY & \xleftarrow{Fg} & FX & \xleftarrow{f_X} & T_S X & & \\
 & & & & \eta_X^s \uparrow & \text{(by def.)} & \\
 & & & & X & & \square
 \end{array}$$

The above result turns out to be a useful tool to easily obtain measure evaluation functions satisfying the naturality and measurability requirements.

Example 17. Consider the measure terms evaluation function for FlatCCS given in Example 8. Here we show how it arises as an instance of structural λ -iteration, for a suitable distributive law of type $\lambda: M(\mathbb{R}^2 \times Id) \Rightarrow (\mathbb{R}^2 \times Id)M$.

Let $\lambda: M(\mathbb{R}^2 \times Id) \Rightarrow (\mathbb{R}^2 \times Id)M$ be, for $x, x' \in X$ and $z, z' \in \mathbb{R}^2$,

$$\begin{aligned}
 \lambda_X((z, x) \blacktriangleleft (z', x')) &= (z, x \blacktriangleleft x') \\
 \lambda_X((z, x) \blacktriangleright (z', x')) &= (z', x \blacktriangleright x') \\
 \lambda_X((z, x) \blacktriangledown (z', x')) &= (\tfrac{1}{2}(z + z'), x \blacktriangledown x') \\
 \lambda_X(\langle (z, x) \rangle_{z'}) &= (z + z', \langle x \rangle_{z'}).
 \end{aligned}$$

Then, define $k: Id \Rightarrow (\mathbb{R}^2 \times Id)$, and a $\varphi: M(\mathbb{R}^2 \times \Delta T_S) \Rightarrow \Delta T_S$ as follows, for $x \in X$, $z, z' \in \mathbb{R}^2$, and $\beta, \beta' \in \Delta T_S X$,

$$\begin{aligned} k_X(x) &= ((0, 0), x), \\ \varphi_X((z, \beta) \blacktriangleleft (z', \beta')) &= (\beta \times \beta') \circ (\lambda(x, x') \cdot x \parallel x')^{-1} \\ \varphi_X((z, \beta) \blacktriangleright (z', \beta')) &= (\beta \times \beta') \circ (\lambda(x, x') \cdot x \parallel x')^{-1} \\ \varphi_X((z, \beta) \blacktriangledown (z', \beta')) &= (e^{-\|z-z'\|} \cdot (\beta \times \beta')) \circ (\lambda(x, x') \cdot x \parallel x')^{-1} \\ \varphi_X(\langle (z, \beta) \rangle_{z'}) &= \beta \circ (\lambda x \cdot [x]_{z'})^{-1}. \end{aligned}$$

These are easily seen to be natural and measurable. Now, applying the structural λ -iteration proof principle of Proposition 15 we obtain

$$\begin{array}{ccccc} M(\mathbb{R}^2 \times \Delta T_S X) & \xleftarrow{M(\mathbb{R}^2 \times \langle \cdot \rangle_X)} & M(\mathbb{R}^2 \times T_M \Delta X) & \xleftarrow{M\langle pos, id \rangle_{\Delta X}} & MT_M \Delta X \\ \varphi_X \downarrow & & \langle \cdot \rangle_X \dashrightarrow & & \downarrow \psi_X \\ \Delta T_S X & & T_M \Delta X & & \Delta X \\ & \swarrow \Delta \eta_X^s & & \nwarrow \eta_X^m & \\ & & & & \\ X & \xrightarrow{\eta_X^m} & T_M X & \xleftarrow{\psi_X} & MT_M X \\ k_X \downarrow & \langle pos, id \rangle_X \downarrow & \downarrow & & \downarrow M\langle pos, id \rangle_X \\ \mathbb{R}^2 \times X & \xrightarrow{(\mathbb{R}^2 \times \eta_X^m)} & \mathbb{R}^2 \times T_M X & \xleftarrow{(\mathbb{R}^2 \times \psi_X) \circ \lambda_{(T_M X)}} & M(\mathbb{R}^2 \times T_M X) \end{array}$$

It is easy to check that the diagrams above commute also when $\langle \cdot \rangle_X^{\text{fl}}$ is used in place of $\langle \cdot \rangle_X$, hence $\langle \cdot \rangle_X^{\text{fl}} = \langle \cdot \rangle_X$. Naturality follows by Corollary 16. \blacksquare

Although we have seen that the proposed λ -iteration proof principle is expressive enough for implementing a “metric” semantics for the FlatCCS (but is not limited to this) we must recall that in [6] there are other and more expressive λ -coiterative proof principles, depending on the type of the distributive law λ at hand. Therefore, this method can be further extended by dualizing [6] in all its variants of coinductive proof principles.

6 Conclusions

In this paper, we have introduced *Measure GSOS*, an SOS specification format for continuous state probabilistic calculi, such that the induced behavioural equivalence is a congruence. As an example, we have introduced *FlatCCS*, a CCS-like probabilistic calculus of processes living on the Euclidean plane.

In this format, transitions have the form $p \xrightarrow{\alpha} \mu$, where μ is a “measure term”, i.e., an expression denoting a probability measure over sets of processes. An MGSOS specification $(\mathcal{R}, \langle \cdot \rangle)$ is composed by a set \mathcal{R} of GSOS-like rules, and a measure terms evaluation function $\langle \cdot \rangle: T_M \Delta \Rightarrow \Delta T_S$. The rule set yields a LTS corresponding to the collective semantics of all the \mathcal{R} -derivable measure terms for a given process. Then, each measure term is given an “interpretation” via $\langle \cdot \rangle$, and the overall probabilistic semantics is given by summing up the set of partial behaviours. MGSOS specification systems are then proved to yield an abstract GSOS distributive law of type $S(Id \times \Delta^L) \Rightarrow (\Delta T_S)^L$.

It is interesting to compare this format with the usual GSOS. In particular, in a transition $p \xrightarrow{\alpha} \mu$, the source term p and the target term μ are from different languages, defined by two different syntactic monads (T_S and T_M respectively). The connection between these two languages is provided by $\langle \cdot \rangle$, which is a kind of “distributive law” across two languages. Notably, the usual GSOS format can be seen as a special case, when $T_S = T_M$ and Δ is not present; in this case, $\langle \cdot \rangle = Id$. Thus, a possible future work is to investigate a more general GSOS-like format, encompassing both MGSOS and GSOS specifications.

Another interesting future work is to extend this development to the case of continuous *stochastic* systems. Basically, we have to replace the probabilistic monad Δ with the stochastic monad G . We need to prove that this monad admits a final coalgebra, but this can be shown by adapting the proof given in [30].

Acknowledgements The authors wish to thank the anonymous referees for the useful comments on the preliminary version of this paper, in particular for suggesting a simpler construction of the initial algebra in the proof of Theorem 4.

References

1. E. A. Abbott. *Flatland: A Romance of Many Dimensions*. Blackwell, 1884.
2. J. Adámek and V. Trnková. *Automata and Algebras in Categories*. Kluwer Academic Publishers, Norwell, MA, USA, 1st edition, 1990.
3. G. Bacci and M. Miculan. Measurable stochastics for Brane Calculus. *Theoretical Comput. Sci.*, 431:117–136, 2012. doi:10.1016/j.tcs.2011.12.055.
4. R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, and G. Pardini. Spatial calculus of looping sequences. *Electr. Notes Theor. Comput. Sci.*, 229(1):21–39, 2009.
5. M. Barr. Algebraically compact functors. *Journal of Pure and Applied Algebra*, 82(3):211 – 231, 1992.
6. F. Bartels. *On Generalised Coinduction and Probabilistic Specification Formats: Distributive Laws in Coalgebraic Modelling*. PhD thesis, CWI, Amsterdam, 2004.
7. F. Bartels, A. Sokolova, and E. P. de Vink. A hierarchy of probabilistic system types. *Electr. Notes Theor. Comput. Sci.*, 82(1), 2003.
8. B. Bloom, S. Istrail, and A. R. Meyer. Bisimulation can’t be traced. *J. ACM*, 42(1):232–268, 1995.
9. L. Brodo, P. Degano, S. Gilmore, J. Hillston, and C. Priami. Performance evaluation for global computation. In C. Priami, editor, *Proc. Global Computing*, volume 2874 of *Lecture Notes in Computer Science*, pages 229–253. Springer, 2003.

10. L. Cardelli and P. Gardner. Processes in space. In F. Ferreira, B. Löwe, E. Mayordomo, and L. M. Gomes, editors, *Proc. CiE*, volume 6158 of *Lecture Notes in Computer Science*, pages 78–87. Springer, 2010.
11. L. Cardelli and R. Mardare. The measurable space of stochastic processes. In *Proc. QEST*, pages 171–180. IEEE Computer Society, 2010.
12. V. Danos, J. Desharnais, F. Laviolette, and P. Panangaden. Bisimulation and cocongruence for probabilistic systems. *Inf. Comput.*, 204(4):503–523, 2006.
13. E. P. de Vink and J. J. M. M. Rutten. Bisimulation for probabilistic transition systems: A coalgebraic approach. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *Proc. ICALP*, volume 1256 of *Lecture Notes in Computer Science*, pages 460–470. Springer, 1997.
14. J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labelled Markov processes. *Inf. Comput.*, 179(2):163–193, 2002.
15. J. Ding and J. Hillston. Structural analysis for stochastic process algebra models. In M. Johnson and D. Pavlovic, editors, *Proc. AMAST*, volume 6486 of *Lecture Notes in Computer Science*, pages 1–27. Springer, 2010.
16. E.-E. Doberkat. *Stochastic relations: foundations for Markov transition systems*. Chapman & Hall/CRC studies in informatics series. Chapman & Hall/CRC, 2007.
17. M. Giry. A categorical approach to probability theory. In Banaschewski, editor, *Categorical Aspects of Topology and Analysis*, volume 915 of *Lecture Notes in Mathematics*, pages 68–85. Springer Berlin/Heidelberg, 1982.
18. H. Hermanns, U. Herzog, and J.-P. Katoen. Process algebra for performance evaluation. *Theor. Comput. Sci.*, 274(1-2):43–87, 2002.
19. J. Hillston. Process algebras for quantitative analysis. In *Proc. LICS*, pages 239–248. IEEE Computer Society, 2005.
20. B. Klin and V. Sassone. Structural operational semantics for stochastic process calculi. In R. M. Amadio, editor, *FoSSaCS*, volume 4962 of *Lecture Notes in Computer Science*, pages 428–442. Springer, 2008.
21. A. Kurz. *Logics for Coalgebras and Applications to Computer Science*. PhD thesis, Ludwig-Maximilians-Universität München, 2000.
22. K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.
23. L. S. Moss and I. D. Viglizzo. Final coalgebras for functors on measurable spaces. *Inf. Comput.*, 204(4):610–636, 2006.
24. P. Panangaden. *Labelled Markov Processes*. Imperial College Press, 2009.
25. G. D. Plotkin. A structural approach to operational semantics. DAIMI FN-19, Computer Science Department, Århus University, Århus, Denmark, 1981.
26. M. B. Smyth and G. D. Plotkin. The category-theoretic solution of recursive domain equations. *SIAM J. Comput.*, 11(4):761–783, 1982.
27. A. Sokolova. Probabilistic systems coalgebraically: A survey. *Theoretical Comput. Sci.*, 412(38):5095–5110, 2011.
28. S. Staton. Relating coalgebraic notions of bisimulation. *Logical Methods in Computer Science*, 7(1), 2011.
29. D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Proc. 12th LICS Conf.*, pages 280–291. IEEE Computer Society Press, 1997.
30. I. D. Viglizzo. Final sequences and final coalgebras for measurable spaces. In J. L. Fiadeiro, N. Harman, M. Roggenbach, and J. J. M. M. Rutten, editors, *Proc. CALCO*, volume 3629 of *Lecture Notes in Computer Science*, pages 395–407. Springer, 2005.