
COMPOSITIONAL BISIMULATION METRIC REASONING WITH PROBABILISTIC PROCESS CALCULI*

DANIEL GEBLER^a, KIM G. LARSEN^b, AND SIMONE TINI^c

^a VU University Amsterdam (NL)
e-mail address: e.d.gebler@vu.nl

^b Aalborg University (DK)
e-mail address: kgl@cs.aau.dk

^c University of Insubria (IT)
e-mail address: simone.tini@uninsubria.it

ABSTRACT. We study which standard operators of probabilistic process calculi allow for compositional reasoning with respect to bisimulation metric semantics. We argue that uniform continuity (generalizing the earlier proposed property of non-expansiveness) captures the essential nature of compositional reasoning and allows now also to reason compositionally about recursive processes. We characterize the distance between probabilistic processes composed by standard process algebra operators. Combining these results, we demonstrate how compositional reasoning about systems specified by continuous process algebra operators allows for metric assume-guarantee like performance validation.

1. INTRODUCTION

Probabilistic process algebras, such as probabilistic CCS [JLY01, Bar04, DD07], CSP [JLY01, Bar04, DvGH⁺07, DL12] and ACP [And99, And02], are languages that are employed to describe probabilistic concurrent communicating systems, or probabilistic processes for short. Nondeterministic probabilistic transition systems [Seg95] combine labeled transition systems [Kel76] and discrete time Markov chains [Ste94, HJ94]. They allow us to model separately the reactive system behavior, nondeterministic choices and probabilistic choices.

Behavioral semantics provide formal notions to compare systems. Behavioral equivalences are behavioral semantics that allow us to determine the observational equivalence of systems by abstracting from behavioral details that may be not relevant in a given application context. In essence, behavioral equivalences equate processes that are indistinguishable to any external observer. The

2012 ACM CCS: [Theory of computation]: Models of Computation—Concurrency—Process Calculi; Semantics and reasoning—Program semantics.

Key words and phrases: probabilistic process algebra, bisimulation metric semantics, compositional reasoning, uniform continuity.

* A preliminary version of this paper appeared as [GLT15].

This research is partially supported by the European FET projects SENSATION and CASSTING and the Sino-Danish Center IDEA4CPS..

most prominent example is bisimulation equivalence [LS91, SL95, Seg95], which provides a well-established theory of the behavior of probabilistic nondeterministic transition systems.

Recently it became clear that the notion of behavioral equivalence is too strict in the context of probabilistic models. The probability values in those models originate either from observations (statistical sampling) or from requirements (probabilistic specification). Behavioral equivalences such as bisimulation equivalence are binary notions that can only answer the question if two systems behave precisely the same way or not. However, a tiny variation of the probabilities, which may be due to a measurement error or limitations how precise a specified probabilistic choice can be realized in a concrete system, will make these systems behaviorally inequivalent without any further information. In practice, many systems are approximately correct. This leads immediately to the question of what is an appropriate notion to measure the quality of the approximation. The most prominent notion is behavioral metric semantics [DGJP04, vBW05, DCP06] which provides a behavioral distance that characterizes how far the behavior of two systems is apart. Bisimulation metrics are the quantitative analogue to bisimulation equivalences and assign to each pair of processes a distance which measures the proximity of their quantitative properties. The distances form a pseudometric¹ with bisimilar processes at distance 0.

In order to specify and verify systems in a compositional manner, it is necessary that the behavioral semantics is compatible with all operators of the language that describe these systems. For behavioral equivalence semantics there is common agreement that compositional reasoning requires that the considered behavioral equivalence is a congruence with respect to all language operators. For example, consider a term $f(s_1, s_2)$ which describes a system consisting of subcomponents s_1 and s_2 that are composed by the binary operator f . When replacing s_1 with a behaviorally equivalent s'_1 , and s_2 with a behaviorally equivalent s'_2 , congruence of the operator f guarantees that the composed system $f(s_1, s_2)$ is behaviorally equivalent to the resulting replacement system $f(s'_1, s'_2)$. This implies that equivalent systems are inter-substitutable: Whenever a system s in a language context $C[s]$ is replaced by an equivalent system s' , the obtained context $C[s']$ is equivalent to $C[s]$. The congruence property is important since it is usually much easier to model and study (a set of) small systems and then combine them together rather than to work with a large monolithic system.

However, for behavioral metric semantics there is no satisfactory understanding of which property an operator should satisfy in order to facilitate compositional reasoning. Intuitively, what is needed is a formalization of the idea that systems close to each other should be approximately inter-substitutable: Whenever a system s in a language context $C[s]$ is replaced by a close system s' , the obtained context $C[s']$ should be close to $C[s]$. In other words, there should be some relation between the behavioral distance between s and s' and the behavioral distance between $C[s]$ and $C[s']$. This ensures that any limited change in the behavior of a subcomponent s implies a smooth and limited change in the behavior of the composed system $C[s]$ (absence of chaotic behavior when system components and parameters are modified in a controlled manner). Earlier proposals such as non-expansiveness [DGJP04] and non-extensiveness [BBLM13] are only partially satisfactory for non-recursive operators and even worse, they do not allow at all to reason compositionally over recursive processes. More fundamentally, those proposals are kind of ‘ad hoc’ and do not capture systematically the essential nature of compositional metric reasoning.

In this paper we consider uniform continuity as a property that generalizes non-extensiveness and non-expansiveness and captures the essential nature of compositional reasoning w.r.t. behavioral metric semantics. A uniformly continuous binary process operator f ensures that for any non-zero

¹A bisimulation metric is in fact a pseudometric. For convenience we use the term bisimulation metric instead of bisimulation pseudometric.

bisimulation distance ϵ (understood as the admissible tolerance from the operational behavior of the composed process $f(s_1, s_2)$) there are non-zero bisimulation distances δ_1 and δ_2 (understood as the admissible tolerances from the operational behavior of the processes s_1 and s_2) such that the distance between the composed processes $f(s_1, s_2)$ and $f(s'_1, s'_2)$ is at most ϵ whenever the component s'_1 (resp. s'_2) is in distance of at most δ_1 from s_1 (resp. at most δ_2 from s_2). Uniform continuity ensures that a small variance in the behavior of the parts leads to a bounded small variance in the behavior of the composed processes. Since uniformly continuous operators preserve the convergence of sequences, this allows us to approximate composed systems by approximating its subsystems. In summary, uniform continuity allows us to investigate the behavior of systems by disassembling them into their components, analyze at the component level, and then derive properties of the composed system. We consider the uniform notion of continuity (technically, the δ_i depend only on ϵ and are independent of the concrete systems s_i) because we aim at universal compositionality guarantees. As important notion of uniform continuity we consider Lipschitz continuity which ensures that the ratio between the distance of composed processes and the distance between its parts is bounded.

Our main contributions are as follows:

- (1) We develop for many non-recursive and recursive process operators used in various probabilistic process algebras tight upper bounds on the distance between processes combined by those operators (Sections 3.2 and 4.2).
- (2) We show that non-recursive process operators, esp. (nondeterministic and probabilistic variants of) sequential, alternative and parallel composition, allow for compositional reasoning w.r.t. the compositionality criteria of non-expansiveness and hence also w.r.t. both Lipschitz and uniform continuity (Section 3).
- (3) We show that recursive process operators, e.g. (nondeterministic and probabilistic variants of) Kleene-star iteration and π -calculus bang replication, allow for compositional reasoning w.r.t. the compositionality criterion of Lipschitz continuity and hence also w.r.t. uniform continuity, but not w.r.t. non-expansiveness and non-extensiveness (Section 4).
- (4) We discuss the copy operator proposed in [BIM95, FvGdW12] to specify the fork operation of operating systems as an example of operator allowing for compositional reasoning w.r.t. the compositionality criterion of uniform continuity, but not w.r.t. Lipschitz continuity.
- (5) We demonstrate the practical relevance of our methods by reasoning compositionally over a network protocol built from uniformly continuous operators. In detail, we show how to derive performance guarantees for the entire system from performance assumptions about individual components. In reverse, we show also how to derive performance requirements on individual components from performance requirements of the complete system (Section 5).

2. PRELIMINARIES

2.1. Probabilistic Transition Systems. We consider transition systems with process terms as states and labeled transitions taking states to distributions over states. Process terms are inductively defined by process combinators.

Definition 2.1 (Signature). A *signature* is a structure $\Sigma = (F, r)$, where

- (1) F is a countable set of *operators*, and
- (2) $r: F \rightarrow \mathbb{N}$ is a *rank function*.

The rank function gives by $r(f)$ the arity of operator f . We call operators with arity 0 *constants*. If the rank of f is clear from the context we will use the symbol n for $r(f)$. We may write $f \in \Sigma$ as shorthand for $\Sigma = (F, r)$ with $f \in F$.

Terms are defined by structural recursion over the signature. We assume an infinite set of *state variables* \mathcal{V}_s disjoint from F .

Definition 2.2 (State terms). The set of *state terms* over a signature Σ and a set $V \subseteq \mathcal{V}_s$ of state variables, notation $\mathbb{T}(\Sigma, V)$, is the least set satisfying:

- $V \subseteq \mathbb{T}(\Sigma, V)$, and
- $f(t_1, \dots, t_n) \in \mathbb{T}(\Sigma, V)$ whenever $f \in \Sigma$ and $t_1, \dots, t_n \in \mathbb{T}(\Sigma, V)$.

We write c for $c()$ if c is a constant. The set of *closed state terms* $\mathbb{T}(\Sigma, \emptyset)$ is abbreviated as $\mathbb{T}(\Sigma)$. The set of *open state terms* $\mathbb{T}(\Sigma, \mathcal{V}_s)$ is abbreviated as $\mathbb{T}(\Sigma)$. We may refer to operators in Σ as *process combinators*, to state variables in \mathcal{V}_s as *process variables*, and to closed state terms in $\mathbb{T}(\Sigma)$ as *processes*.

A probability distribution over the set of closed state terms $\mathbb{T}(\Sigma)$ is a mapping $\pi: \mathbb{T}(\Sigma) \rightarrow [0, 1]$ with $\sum_{t \in \mathbb{T}(\Sigma)} \pi(t) = 1$ that assigns to each closed term $t \in \mathbb{T}(\Sigma)$ its respective probability $\pi(t)$. The probability mass of a set of closed terms $T \subseteq \mathbb{T}(\Sigma)$ in some probability distribution π is given by $\pi(T) = \sum_{t \in T} \pi(t)$. We denote by $\Delta(\mathbb{T}(\Sigma))$ the set of all probability distributions over $\mathbb{T}(\Sigma)$. We let π, π' range over $\Delta(\mathbb{T}(\Sigma))$.

Notation 2.3 (Notations for probability distributions). We denote by $\delta(t)$ with $t \in \mathbb{T}(\Sigma)$ the *Dirac distribution* defined by $(\delta(t))(t) = 1$ and $(\delta(t))(t') = 0$ for all $t' \in \mathbb{T}(\Sigma)$ with $t \neq t'$. The convex combination $\sum_{i \in I} p_i \pi_i$ of a family $\{\pi_i\}_{i \in I}$ of probability distributions $\pi_i \in \Delta(\mathbb{T}(\Sigma))$ with $p_i \in (0, 1]$ and $\sum_{i \in I} p_i = 1$ is defined by $(\sum_{i \in I} p_i \pi_i)(t) = \sum_{i \in I} (p_i \pi_i(t))$ for all terms $t \in \mathbb{T}(\Sigma)$. The expression $f(\pi_1, \dots, \pi_n)$ with $f \in \Sigma$ and $\pi_i \in \Delta(\mathbb{T}(\Sigma))$ denotes the product distribution of π_1, \dots, π_n defined by $(f(\pi_1, \dots, \pi_n))(f(t_1, \dots, t_n)) = \prod_{i=1}^n \pi_i(t_i)$ and $(f(\pi_1, \dots, \pi_n))(t) = 0$ for all terms $t \in \mathbb{T}(\Sigma)$ not in the form $t = f(t_1, \dots, t_n)$. For binary operators f we may use the infix notation and write $\pi_1 f \pi_2$ for $f(\pi_1, \pi_2)$.

Next, we introduce a language to describe probability distributions. We assume an infinite set of *distribution variables* \mathcal{V}_d and let μ, ν range over \mathcal{V}_d . We denote by \mathcal{V} the set of state and distribution variables $\mathcal{V} = \mathcal{V}_s \cup \mathcal{V}_d$ and let ζ, ζ' range over \mathcal{V} .

Definition 2.4 (Distribution terms). The set of *distribution terms* over a signature Σ , a set of state variables $V_s \subseteq \mathcal{V}_s$ and a set of distribution variables $V_d \subseteq \mathcal{V}_d$, notation $\text{DT}(\Sigma, V_s, V_d)$, is the least set satisfying:

- (1) $V_d \subseteq \text{DT}(\Sigma, V_s, V_d)$,
- (2) $\{\delta(t) \mid t \in \mathbb{T}(\Sigma, V_s)\} \subseteq \text{DT}(\Sigma, V_s, V_d)$,
- (3) $\sum_{i \in I} p_i \theta_i \in \text{DT}(\Sigma, V_s, V_d)$ whenever $\theta_i \in \text{DT}(\Sigma, V_s, V_d)$ and $p_i \in (0, 1]$ with $\sum_{i \in I} p_i = 1$, and
- (4) $f(\theta_1, \dots, \theta_n) \in \text{DT}(\Sigma, V_s, V_d)$ whenever $f \in \Sigma$ and $\theta_1, \dots, \theta_n \in \text{DT}(\Sigma, V_s, V_d)$.

Distribution terms have the following meaning. A *distribution variable* $\mu \in \mathcal{V}_d$ is a variable that takes values from $\Delta(\mathbb{T}(\Sigma))$. An *instantiable Dirac distribution* $\delta(t)$ is an expression that takes as value the Dirac distribution $\delta(t')$ when state variables in t are substituted such that t becomes the closed term t' . Case 3 allows us to construct convex combinations of distributions. Case 4 lifts structural recursion from state terms to distribution terms.

The set of *closed distribution terms* $\text{DT}(\Sigma, \emptyset, \emptyset)$ is abbreviated as $\text{DT}(\Sigma)$. The set of *open distribution terms* $\text{DT}(\Sigma, \mathcal{V}_s, \mathcal{V}_d)$ is abbreviated as $\mathbb{DT}(\Sigma)$. We write $\theta_1 \oplus_p \theta_2$ for $\sum_{i=1}^2 p_i \theta_i$ with $p_1 = p$ and $p_2 = 1 - p$. Furthermore, for binary operators f we may use the infix notation and write $\theta_1 f \theta_2$ for $f(\theta_1, \theta_2)$.

Definition 2.5 (Substitution). A *substitution* is a mapping $\sigma: \mathcal{V} \rightarrow \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)$ such that $\sigma(x) \in \mathbb{T}(\Sigma)$, if $x \in \mathcal{V}_s$, and $\sigma(\mu) \in \mathbb{DT}(\Sigma)$, if $\mu \in \mathcal{V}_d$. A substitution σ extends to a mapping from state terms to state terms by $\sigma(f(t_1, \dots, t_n)) = f(\sigma(t_1), \dots, \sigma(t_n))$. A substitution σ extends to a mapping from distribution terms to distribution terms by

- (i) $\sigma(\delta(t)) = \delta(\sigma(t))$,
- (ii) $\sigma(\sum_{i \in I} p_i \theta_i) = \sum_{i \in I} p_i \sigma(\theta_i)$, and
- (iii) $\sigma(f(\theta_1, \dots, \theta_n)) = f(\sigma(\theta_1), \dots, \sigma(\theta_n))$.

A substitution σ is *closed* if $\sigma(x) \in \mathbb{T}(\Sigma)$ for all $x \in \mathcal{V}_s$ and $\sigma(\mu) \in \mathbb{DT}(\Sigma)$ for all $\mu \in \mathcal{V}_d$. Notice that closed distribution terms denote distributions in $\Delta(\mathbb{T}(\Sigma))$.

Probabilistic nondeterministic labelled transition systems [Seg95], PTSs for short, extend labelled transition systems by allowing for probabilistic choices in the transitions. As state space we will take the set of all closed terms $\mathbb{T}(\Sigma)$.

Definition 2.6 (PTS, [Seg95]). A *probabilistic nondeterministic labeled transition system (PTS)* over the signature Σ is given by a triple $(\mathbb{T}(\Sigma), A, \rightarrow)$, where:

- $\mathbb{T}(\Sigma)$ is the set of all closed terms over Σ ,
- A is a countable set of *actions*, and
- $\rightarrow \subseteq \mathbb{T}(\Sigma) \times A \times \Delta(\mathbb{T}(\Sigma))$ is a *transition relation*.

We call $(t, a, \pi) \in \rightarrow$ a *transition* from state t to distribution π labelled by action a . We write $t \xrightarrow{a} \pi$ for $(t, a, \pi) \in \rightarrow$. Moreover, we write $t \xrightarrow{a}$ if there exists some distribution $\pi \in \Delta(\mathbb{T}(\Sigma))$ with $t \xrightarrow{a} \pi$, and $t \not\xrightarrow{a}$ if there is no distribution $\pi \in \Delta(\mathbb{T}(\Sigma))$ with $t \xrightarrow{a} \pi$. For a closed term $t \in \mathbb{T}(\Sigma)$ and an action $a \in A$, let $\text{der}(t, a) = \{\pi \in \Delta(\mathbb{T}(\Sigma)) \mid t \xrightarrow{a} \pi\}$ denote the set of all distributions reachable from t by performing an a -labeled transition. We call $\text{der}(t, a)$ also the *a -derivatives* of t .

We say that a PTS is *image-finite* if $\text{der}(t, a)$ is finite for each closed term t and action a . In the rest of the paper we assume to deal with image finite PTSs.

2.2. Bisimulation metric. Bisimulation metric² [DGJP04, vBW05, DCP06] provides a robust semantics for PTSs. It is the quantitative analogue to bisimulation equivalence and assigns to each pair of states a distance which measures the proximity of their quantitative properties. The distances form a pseudometric where bisimilar processes are at distance 0.

Definition 2.7 (Pseudometric over $\mathbb{T}(\Sigma)$). A function $d: \mathbb{T}(\Sigma) \times \mathbb{T}(\Sigma) \rightarrow [0, 1]$ is a *1-bounded pseudometric* if

- $d(t, t) = 0$ for all $t \in \mathbb{T}(\Sigma)$,
- $d(t, t') = d(t', t)$ for all $t, t' \in \mathbb{T}(\Sigma)$ (symmetry), and
- $d(t, t') \leq d(t, t'') + d(t'', t')$ for all $t, t', t'' \in \mathbb{T}(\Sigma)$ (triangle inequality).

We will define later bisimulation metrics as 1-bounded pseudometrics that measure how much two states disagree on their reactive behavior and their probabilistic choices. Note that a pseudometric d permits that $d(t, t') = 0$ even if t and t' are different terms (in contrast to a metric d). This will allow us to assign distance 0 to different bisimilar states. We will provide two (equivalent) characterizations of bisimulation metrics in terms of a coinductive definition pattern and in terms of fixed points.

²A bisimulation metric is in fact a pseudometric. In line with the literature we use the term bisimulation metric instead of bisimulation pseudometric.

Both characterizations require the following lattice structure. Let $([0, 1]^{\mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma)}, \sqsubseteq)$ be the complete lattice of functions $d: \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \rightarrow [0, 1]$ ordered by $d_1 \sqsubseteq d_2$ iff $d_1(t, t') \leq d_2(t, t')$ for all $t, t' \in \mathsf{T}(\Sigma)$. Then for each $D \subseteq [0, 1]^{\mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma)}$ the supremum and infimum are $\sup(D)(t, t') = \sup_{d \in D} d(t, t')$ and $\inf(D)(t, t') = \inf_{d \in D} d(t, t')$ for all $t, t' \in \mathsf{T}(\Sigma)$. The bottom element is the constant zero function $\mathbf{0}$ given by $\mathbf{0}(t, t') = 0$, and the top element is the constant one function $\mathbf{1}$ given by $\mathbf{1}(t, t') = 1$, for all $t, t' \in \mathsf{T}(\Sigma)$.

2.2.1. Metrical lifting. Bisimulation metric is characterized using the quantitative analogous of the bisimulation game, meaning that two states $t, t' \in \mathsf{T}(\Sigma)$ at some given distance can mimic each other's transitions and evolve to distributions that are at distance not greater than the distance between the source states. Technically, we need a notion that lifts pseudometrics from states to distributions (to capture probabilistic choices).

A 1-bounded pseudometric on terms $\mathsf{T}(\Sigma)$ is lifted to a 1-bounded pseudometric on distributions $\Delta(\mathsf{T}(\Sigma))$ by means of the Kantorovich pseudometric [DD09]. This lifting is the quantitative analogous of the lifting of bisimulation equivalence relations on terms to bisimulation equivalence relations on distributions [vBW01].

A *matching* for a pair of distributions $(\pi, \pi') \in \Delta(\mathsf{T}(\Sigma)) \times \Delta(\mathsf{T}(\Sigma))$ is a distribution over the product state space $\omega \in \Delta(\mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma))$ with left marginal π , i.e. $\sum_{t' \in \mathsf{T}(\Sigma)} \omega(t, t') = \pi(t)$ for all $t \in \mathsf{T}(\Sigma)$, and right marginal π' , i.e. $\sum_{t \in \mathsf{T}(\Sigma)} \omega(t, t') = \pi'(t')$ for all $t' \in \mathsf{T}(\Sigma)$. Let $\Omega(\pi, \pi')$ denote the set of all matchings for (π, π') . Intuitively, a matching $\omega \in \Omega(\pi, \pi')$ may be understood as a transportation schedule that describes the shipment of probability mass from π to π' . Historically this motivation dates back to the Monge-Kantorovich optimal transport problem [Vil08].

Definition 2.8 (Kantorovich lifting). Let $d: \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \rightarrow [0, 1]$ be a 1-bounded pseudometric. The *Kantorovich lifting* of d is a 1-bounded pseudometric $\mathbf{K}(d): \Delta(\mathsf{T}(\Sigma)) \times \Delta(\mathsf{T}(\Sigma)) \rightarrow [0, 1]$ defined by

$$\mathbf{K}(d)(\pi, \pi') = \min_{\omega \in \Omega(\pi, \pi')} \sum_{t, t' \in \mathsf{T}(\Sigma)} d(t, t') \cdot \omega(t, t')$$

for all $\pi, \pi' \in \Delta(\mathsf{T}(\Sigma))$. We call $\mathbf{K}(d)$ the *Kantorovich pseudometric* of d .

In order to capture nondeterministic choices, we need to lift pseudometrics on distributions to pseudometrics on sets of distributions.

Definition 2.9 (Hausdorff lifting). Let $\hat{d}: \Delta(\mathsf{T}(\Sigma)) \times \Delta(\mathsf{T}(\Sigma)) \rightarrow [0, 1]$ be a 1-bounded pseudometric. The *Hausdorff lifting* of \hat{d} is a 1-bounded pseudometric $\mathbf{H}(\hat{d}): P(\Delta(\mathsf{T}(\Sigma))) \times P(\Delta(\mathsf{T}(\Sigma))) \rightarrow [0, 1]$ defined by

$$\mathbf{H}(\hat{d})(\Pi_1, \Pi_2) = \max \left\{ \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \hat{d}(\pi_1, \pi_2), \sup_{\pi_2 \in \Pi_2} \inf_{\pi_1 \in \Pi_1} \hat{d}(\pi_2, \pi_1) \right\}$$

for all $\Pi_1, \Pi_2 \subseteq \Delta(\mathsf{T}(\Sigma))$, with $\inf \emptyset = 1$, and $\sup \emptyset = 0$. We call $\mathbf{H}(\hat{d})$ the *Hausdorff pseudometric* of \hat{d} .

2.2.2. Coinductive characterization. A 1-bounded pseudometric is a bisimulation metric if for all pairs of terms t and t' each transition of t can be mimicked by a transition of t' with the same label and the distance between the accessible distributions does not exceed the distance between t and t' . By means of a *discount factor* $\lambda \in (0, 1]$, we allow to specify how much the behavioral distance of future transitions is taken into account [DAH03, DGJP04]. The discount factor $\lambda = 1$ expresses

no discount, meaning that the differences in the behavior between t and t' are considered irrespective of after how many steps they can be observed.

Definition 2.10 (Bisimulation metric [DGJP04]). A 1-bounded pseudometric $d: \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \rightarrow [0, 1]$ is a λ -bisimulation metric with $\lambda \in (0, 1]$ if for all terms $t, t' \in \mathsf{T}(\Sigma)$ with $d(t, t') < 1$, if $t \xrightarrow{a} \pi$ then there exists a transition $t' \xrightarrow{a} \pi'$ for a distribution $\pi' \in \Delta(\mathsf{T}(\Sigma))$ such that $\lambda \cdot \mathbf{K}(d)(\pi, \pi') \leq d(t, t')$.

We refer to $\lambda \cdot \mathbf{K}(d)(\pi, \pi') \leq d(t, t')$ as the bisimulation transfer condition. We call the smallest (w.r.t. \sqsubseteq) λ -bisimulation metric λ -bisimilarity metric [DCPP06] and denote it by the symbol \mathbf{d} . We mean by λ -bisimulation distance between t and t' the distance $\mathbf{d}(t, t')$. If λ is clear from the context, we may refer by bisimulation metric, bisimilarity metric and bisimulation distance to λ -bisimulation metric, λ -bisimilarity metric and λ -bisimulation distance. Moreover, we may call the 1-bisimilarity metric also non-discounting bisimilarity metric. Bisimilarity equivalence is the kernel of the λ -bisimilarity metric [DGJP04], namely $\mathbf{d}(t, t') = 0$ iff t and t' are bisimilar.

Example 2.11. Assume a PTS with transitions $\rightarrow = \{s \xrightarrow{a} \pi_s, t \xrightarrow{a} \pi_t\}$ whereby $\pi_s = 0.5\delta(s) + 0.5\delta(0)$ and $\pi_t = (0.5 + \epsilon)\delta(s) + (0.5 - \epsilon)\delta(0)$ for some arbitrary $\epsilon \in [0, 0.5]$. Furthermore, assume a 1-bounded pseudometric d with $d(s, s) = d(0, 0) = 0$ and $d(s, 0) = d(0, s) = 1$. We have $\mathbf{K}(d)(\pi_s, \pi_t) = \epsilon$, by the matching $\omega \in \Omega(\pi_s, \pi_t)$ defined by $\omega(s, s) = 0.5$, $\omega(0, s) = \epsilon$ and $\omega(0, 0) = 0.5 - \epsilon$. Then, d is a bisimulation metric if it satisfies the bisimulation transfer condition $d(s, t) \geq \lambda \mathbf{K}(d)(\pi_s, \pi_t) = \lambda\epsilon$. Moreover, the bisimilarity metric assigns the distance $\mathbf{d}(t, s) = \lambda\epsilon$.

2.2.3. Fixed point characterization. We provide now an alternative characterization of bisimulation metric in terms of prefixed points of an appropriate monotone bisimulation functional [DCPP06]. Bisimilarity metric is then the least fixed point of this functional. Moreover, the fixed point approach allows us also to express up-to- k bisimulation metrics which measure the bisimulation distance for only the first k transition steps.

Definition 2.12 (Bisimulation metric functional). Let $\mathbf{B}: [0, 1]^{\mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma)} \rightarrow [0, 1]^{\mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma)}$ be the function defined by

$$\mathbf{B}(d)(t, t') = \sup_{a \in A} \{\mathbf{H}(\lambda \cdot \mathbf{K}(d))(der(t, a), der(t', a))\}$$

for $d: \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \rightarrow [0, 1]$ and $t, t' \in \mathsf{T}(\Sigma)$, with $(\lambda \cdot \mathbf{K}(d))(\pi, \pi') = \lambda \cdot \mathbf{K}(d)(\pi, \pi')$.

It is easy to show that \mathbf{B} is a monotone function on $([0, 1]^{\mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma)}, \sqsubseteq)$. The following Proposition characterizes bisimulation metrics as prefixed points of \mathbf{B} .

Proposition 2.13 ([DCPP06]). *Let $d: \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \rightarrow [0, 1]$ be a 1-bounded pseudometric. Then $\mathbf{B}(d) \sqsubseteq d$ iff d is a bisimulation metric.*

Proposition 2.13 provides the fixed point characterization of bisimulation metrics and shows that it coincides with the coinductive characterization of Definition 2.10. Since \mathbf{B} is a monotone function on the complete lattice $([0, 1]^{\mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma)}, \sqsubseteq)$, we can characterize the bisimilarity metric as least fixed point of \mathbf{B} .

Proposition 2.14 ([DCPP06]). *The bisimilarity metric \mathbf{d} is the least fixed point of \mathbf{B} .*

Moreover, the fixed point approach allows us to define a notion of bisimulation distance that considers only the first k transition steps.

Definition 2.15 (Up-to- k bisimilarity metric). We define the up-to- k bisimilarity metric \mathbf{d}_k for $k \in \mathbb{N}$ by $\mathbf{d}_k = \mathbf{B}^k(\mathbf{0})$.

We call $\mathbf{d}_k(s, t)$ the up-to- k bisimulation distance between s and t .

If the PTS is image-finite and, moreover, for each transition $t \xrightarrow{a} \pi$ we have that the support of π is finite, then \mathbf{B} is monotone and continuous, which ensures that the closure ordinal of \mathbf{B} is ω [vB12]-Section 3. As a consequence, up-to- k bisimulation distances converge to the bisimulation distances when $k \rightarrow \infty$, which opens the door to show properties of the bisimulation metric by using a simple inductive argument [vB12].

Proposition 2.16 ([vB12]). *Assume an image-finite PTS s.t. for each transition $t \xrightarrow{a} \pi$ we have that the distribution π has finite support. Then $\mathbf{d} = \lim_{k \rightarrow \infty} \mathbf{d}_k$.*

2.2.4. Properties of bisimulation metrics. We give now an important property of bisimulation metrics that will be essential for the argumentation later in the technical sections.

The bisimulation distance between states t and t' measures the difference of the reactive behavior of t and t' (i.e. which actions can or cannot be performed) along their evolution. An important distinction is if two states can perform the same initial actions. In this case, the behavioral distance is given by the bisimulation game on the derivatives. Otherwise, the two states get the maximal distance of 1 assigned since there is a transition by one of these states that cannot be mimicked by the other state.

We say that states t and t' *do not totally disagree* if $\mathbf{d}(t, t') < 1$. If states do not totally disagree, then they agree on which actions they can perform immediately.

Proposition 2.17. *Let $d: T(\Sigma) \times T(\Sigma) \rightarrow [0, 1]$ be a 1-bounded pseudometric. Then*

- (1) $\mathbf{B}(d)(t, t') < 1$ implies $t \xrightarrow{a} \Leftrightarrow t' \xrightarrow{a}$ for all $a \in A$,
- (2) $d(t, t') < 1$ implies $t \xrightarrow{a} \Leftrightarrow t' \xrightarrow{a}$ for all $a \in A$, if d is a bisimulation metric.

Proof. We start with Proposition 2.17.1 and reason as follows.

$$\begin{aligned} & \mathbf{B}(d)(t, t') < 1 \\ \Leftrightarrow & \forall a \in A. \mathbf{H}(\lambda \cdot \mathbf{K}(d))(der(t, a), der(t', a)) < 1 \\ \Rightarrow & \forall a \in A. ((der(t, a) = \emptyset = der(t', a)) \vee (der(t, a) \neq \emptyset \neq der(t', a))) \\ \Leftrightarrow & \forall a \in A. (t \xrightarrow{a} \Leftrightarrow t' \xrightarrow{a}). \end{aligned}$$

Now we show Proposition 2.17.2. By Proposition 2.13 we get that $d(t, t') < 1$ implies $\mathbf{B}(d)(t, t') < 1$. The thesis follows now from Proposition 2.17.1. \square

Moreover, if $\lambda < 1$ the implications in both cases also hold in the other direction.

Remark 2.18. The bisimulation distance $\mathbf{d}(t, t')$ between terms t and t' is in $[0, \lambda] \cup \{1\}$. If $\lambda \in (0, 1)$, then:

- (1) $\mathbf{d}(t, t') = 1$ iff t can perform an action which t' cannot (or vice versa), i.e. $der(t, a) \neq \emptyset$ and $der(t', a) = \emptyset$ for some action $a \in A$;
- (2) $\mathbf{d}(t, t') = 0$ iff t and t' have the same reactive behavior (are bisimilar); and
- (3) $\mathbf{d}(t, t') \in (0, \lambda]$ iff t and t' have the same set of initial moves, i.e. $der(t, a) = der(t', a)$, and have different reactive behavior after performing the same initial actions.

Notice that in the first case the discount λ does not apply since the different behaviors are observed immediately. If $\lambda = 1$ then the first and last case collapse, i.e. $\mathbf{d}(t, t') = 0$ iff t and t' have the same reactive behavior (are bisimilar), and $\mathbf{d}(t, t') \in (0, 1]$ iff t and t' have different reactive behavior.

2.2.5. *Properties of the Kantorovich lifting.* The Kantorovich pseudometric satisfies important properties that will be essential to prove our technical results. In detail, the Kantorovich lifting functional is monotone, the Dirac operator is an isometric embedding of the metric space of states into the metric space of distributions, and probabilistic choice distributes over the Kantorovich lifting.

Proposition 2.19 ([Pan09]). *Let d and d' be any 1-bounded pseudometrics. Then*

- (1) $\mathbf{K}(d) \subseteq \mathbf{K}(d')$ if $d \subseteq d'$;
- (2) $\mathbf{K}(d)(\delta(t), \delta(t')) = d(t, t')$ for all $t, t' \in \mathcal{T}(\Sigma)$;
- (3) $\mathbf{K}(d)(\sum_{i \in I} p_i \pi_i, \sum_{i \in I} p_i \pi'_i) \leq \sum_{i \in I} p_i \cdot \mathbf{K}(d)(\pi_i, \pi'_i)$ for all $\pi_i, \pi'_i \in \Delta(\mathcal{T}(\Sigma))$ and $p_i \in [0, 1]$ with $\sum_{i \in I} p_i = 1$.

Now we will show a very important new result stating that the Kantorovich lifting preserves concave moduli of continuity of language operators. In other words, moduli of continuity of language operators distribute over probabilistic choices.

Theorem 2.20. *Let $d: \mathcal{T}(\Sigma) \times \mathcal{T}(\Sigma) \rightarrow [0, 1]$ be any 1-bounded pseudometric. Assume an n -ary operator $f \in \Sigma$ and a concave³ function $z: [0, 1]^n \rightarrow [0, 1]$ with*

$$d(f(t_1, \dots, t_n), f(t'_1, \dots, t'_n)) \leq z(d(t_1, t'_1), \dots, d(t_n, t'_n))$$

for all terms $t_1, t'_1, \dots, t_n, t'_n \in \mathcal{T}(\Sigma)$. Then we have

$$\mathbf{K}(d)(f(\pi_1, \dots, \pi_n), f(\pi'_1, \dots, \pi'_n)) \leq z(\mathbf{K}(d)(\pi_1, \pi'_1), \dots, \mathbf{K}(d)(\pi_n, \pi'_n))$$

for all probability distributions $\pi_1, \pi'_1, \dots, \pi_n, \pi'_n \in \Delta(\mathcal{T}(\Sigma))$.

Proof. We assume $\omega_i \in \Omega(\pi_i, \pi'_i)$ to be an optimal matching such that $\mathbf{K}(d)(\pi_i, \pi'_i) = \sum_{t, t' \in \mathcal{T}(\Sigma)} d(t, t') \cdot \omega_i(t, t')$, i.e. a matching between π_i and π'_i which yields the Kantorovich distance $\mathbf{K}(d)(\pi_i, \pi'_i)$. We define a new distribution over the product space $\omega \in \Delta(\mathcal{T}(\Sigma) \times \mathcal{T}(\Sigma))$ by

$$\omega(f(t_1, \dots, t_n), f(t'_1, \dots, t'_n)) = \prod_{i=1}^n \omega_i(t_i, t'_i)$$

for all $t_1, t'_1, \dots, t_n, t'_n \in \mathcal{T}(\Sigma)$. First, we show that ω is a joint probability distribution with left marginal $f(\pi_1, \dots, \pi_n)$ and right marginal $f(\pi'_1, \dots, \pi'_n)$. The left marginal is

$$\begin{aligned} & \sum_{t' \in \mathcal{T}(\Sigma)} \omega(f(t_1, \dots, t_n), t') \\ &= \sum_{t'_1, \dots, t'_n \in \mathcal{T}(\Sigma)} \omega(f(t_1, \dots, t_n), f(t'_1, \dots, t'_n)) \\ &= \sum_{t'_1, \dots, t'_n \in \mathcal{T}(\Sigma)} \prod_{i=1}^n \omega_i(t_i, t'_i) \\ &= \prod_{i=1}^n \sum_{t'_i \in \mathcal{T}(\Sigma)} \omega_i(t_i, t'_i) \\ &= \prod_{i=1}^n \pi_i(t_i) \\ &= f(\pi_1, \dots, \pi_n)(f(t_1, \dots, t_n)) \end{aligned}$$

³A function $z: [0, 1]^n \rightarrow [0, 1]$ is called concave if, for any $x_1, \dots, x_n, y_1, \dots, y_n \in [0, 1]$ and any $\lambda \in [0, 1]$, $z((1-\lambda)x_1 + \lambda y_1, \dots, (1-\lambda)x_n + \lambda y_n) \geq (1-\lambda)z(x_1, \dots, x_n) + \lambda z(y_1, \dots, y_n)$.

with $\sum_{t'_1, \dots, t'_n \in \mathbb{T}(\Sigma)} \prod_{i=1}^n \omega_i(t_i, t'_i) = \prod_{i=1}^n \sum_{t'_i \in \mathbb{T}(\Sigma)} \omega_i(t_i, t'_i)$ by induction over n with induction step

$$\begin{aligned}
& \sum_{t'_1, \dots, t'_{n+1} \in \mathbb{T}(\Sigma)} \prod_{i=1}^{n+1} \omega_i(t_i, t'_i) \\
&= \sum_{t'_1, \dots, t'_n \in \mathbb{T}(\Sigma)} \sum_{t'_{n+1} \in \mathbb{T}(\Sigma)} \omega_{n+1}(t_{n+1}, t'_{n+1}) \prod_{i=1}^n \omega_i(t_i, t'_i) \\
&= \sum_{t'_{n+1} \in \mathbb{T}(\Sigma)} \omega_{n+1}(t_{n+1}, t'_{n+1}) \sum_{t'_1, \dots, t'_n \in \mathbb{T}(\Sigma)} \prod_{i=1}^n \omega_i(t_i, t'_i) \\
&= \sum_{t'_{n+1} \in \mathbb{T}(\Sigma)} \omega_{n+1}(t_{n+1}, t'_{n+1}) \prod_{i=1}^n \sum_{t'_i \in \mathbb{T}(\Sigma)} \omega_i(t_i, t'_i) \\
&= \prod_{i=1}^{n+1} \sum_{t'_i \in \mathbb{T}(\Sigma)} \omega_i(t_i, t'_i).
\end{aligned}$$

The right marginal is computed analogously. Hence, $\omega \in \Omega(f(\pi_1, \dots, \pi_n), f(\pi'_1, \dots, \pi'_n))$, i.e. ω is a matching for distributions $f(\pi_1, \dots, \pi_n)$ and $f(\pi'_1, \dots, \pi'_n)$.

The proof obligation can be derived now by

$$\begin{aligned}
& \mathbf{K}(d)(f(\pi_1, \dots, \pi_n), f(\pi'_1, \dots, \pi'_n)) \\
&\leq \sum_{\substack{t_1, \dots, t_n \\ t'_1, \dots, t'_n \in \mathbb{T}(\Sigma)}} d(f(t_1, \dots, t_n), f(t'_1, \dots, t'_n)) \cdot \omega(f(t_1, \dots, t_n), f(t'_1, \dots, t'_n)) \\
&= \sum_{\substack{t_1, \dots, t_n \\ t'_1, \dots, t'_n \in \mathbb{T}(\Sigma)}} d(f(t_1, \dots, t_n), f(t'_1, \dots, t'_n)) \cdot \prod_{i=1}^n \omega_i(t_i, t'_i) \\
&\leq \sum_{\substack{t_1, \dots, t_n \\ t'_1, \dots, t'_n \in \mathbb{T}(\Sigma)}} z(d(t_1, t'_1), \dots, d(t_n, t'_n)) \cdot \prod_{i=1}^n \omega_i(t_i, t'_i) \\
&\leq z \left(\sum_{\substack{t_1, \dots, t_n \\ t'_1, \dots, t'_n \in \mathbb{T}(\Sigma)}} (d(t_1, t'_1), \dots, d(t_n, t'_n)) \cdot \prod_{i=1}^n \omega_i(t_i, t'_i) \right) \\
&= z \left(\sum_{\substack{t_1, \dots, t_n \\ t'_1, \dots, t'_n \in \mathbb{T}(\Sigma)}} \left(d(t_1, t'_1) \cdot \prod_{i=1}^n \omega_i(t_i, t'_i), \dots, d(t_n, t'_n) \cdot \prod_{i=1}^n \omega_i(t_i, t'_i) \right) \right) \\
&= z \left(\left(\sum_{\substack{t_1, \dots, t_n \\ t'_1, \dots, t'_n \in \mathbb{T}(\Sigma)}} d(t_1, t'_1) \cdot \prod_{i=1}^n \omega_i(t_i, t'_i), \dots, \sum_{\substack{t_1, \dots, t_n \\ t'_1, \dots, t'_n \in \mathbb{T}(\Sigma)}} d(t_n, t'_n) \cdot \prod_{i=1}^n \omega_i(t_i, t'_i) \right) \right) \\
&= z \left(\left(\sum_{t_1, t'_1 \in \mathbb{T}(\Sigma)} d(t_1, t'_1) \omega_1(t_1, t'_1), \dots, \sum_{t_n, t'_n \in \mathbb{T}(\Sigma)} d(t_n, t'_n) \omega_n(t_n, t'_n) \right) \right)
\end{aligned}$$

$$= z(\mathbf{K}(d)(\pi_1, \pi'_1), \dots, \mathbf{K}(d)(\pi_n, \pi'_n))$$

whereby the reasoning steps are derived as follows: step 1 from the fact that ω is a matching for distributions $f(\pi_1, \dots, \pi_n)$ and $f(\pi'_1, \dots, \pi'_n)$, step 2 by the definition of ω , step 3 by the assumption $d(f(t_1, \dots, t_n), f(t'_1, \dots, t'_n)) \leq z(d(t_1, t'_1), \dots, d(t_n, t'_n))$, step 4 by using Jensen's inequality for the concave function z , step 7 by $\sum_{t'_1, \dots, t'_n \in \mathbb{T}(\Sigma)} d(t_1, t'_1) \cdot \prod_{i=1}^n \omega_i(t_i, t'_i) = \sum_{t_1, t'_1 \in \mathbb{T}(\Sigma)} d(t_1, t'_1) \omega_1(t_1, t'_1)$, and step 8 by the definition of \mathbf{K} . \square

2.3. PGSOS Specifications. We will specify the operational semantics of operators by SOS rules in the probabilistic GSOS format [Bar04, LGD12, DGL15]. The probabilistic GSOS format, PGSOS format for short, is the quantitative generalization of the classical nondeterministic GSOS format [BIM95]. It is more general than earlier formats [LT05, LT09] which consider transitions of the form $t \xrightarrow{a,q} t'$ modeling that term t reaches through action a the term t' with probability q . The probabilistic GSOS format allows us to specify probabilistic nondeterministic process algebras, such as probabilistic CCS [JLY01, Bar04, DD07], probabilistic CSP [JLY01, Bar04, DvGH⁺07, DL12] and probabilistic ACP [And99, And02].

Definition 2.21 (PGSOS rule, [Bar04, LGD12]). A PGSOS rule r has the form:

$$\frac{\{x_i \xrightarrow{a_{i,k}} \mu_{i,k} \mid i \in I, k \in K_i\} \quad \{x_i \xrightarrow{b_{i,l}} \mid i \in I, l \in L_i\}}{f(x_1, \dots, x_n) \xrightarrow{a} \theta}$$

with $f \in \Sigma$ an operator with rank n , $I = \{1, \dots, n\}$ indices for the arguments of f , K_i, L_i finite index sets, $a_{i,k}, b_{i,l}, a \in A$ actions, $x_i \in \mathcal{V}_s$ state variables, $\mu_{i,k} \in \mathcal{V}_d$ distribution variables, and $\theta \in \mathbb{DT}(\Sigma)$ a distribution term. Furthermore, the following constraints need to be satisfied:

- (1) all $\mu_{i,k}$ for $i \in I, k \in K_i$ are pairwise different;
- (2) all x_1, \dots, x_n are pairwise different;
- (3) $\text{Var}(\theta) \subseteq \{\mu_{i,k} \mid i \in I, k \in K_i\} \cup \{x_1, \dots, x_n\}$.

The PGSOS constraints 1–3 are precisely the constraints of the nondeterministic GSOS format [BIM95] where the variables in the right-hand side of the literals are replaced by distribution variables.

Notation 2.22 (Notations for rules). Let r be a PGSOS rule. The expressions $x_i \xrightarrow{a_{i,k}} \mu_{i,k}$, $x_i \xrightarrow{b_{i,l}}$ and $f(x_1, \dots, x_n) \xrightarrow{a} \theta$ are called, resp., *positive premises*, *negative premises* and *conclusion*. The set of all premises is denoted by $\text{prem}(r)$ and the conclusion by $\text{conc}(r)$. The term $f(x_1, \dots, x_n)$ is called the *source*, the variables x_1, \dots, x_n are called *source variables*, and the distribution term θ is called the *target*.

Given a set of rules R we denote by R_f the rules specifying operator f , i.e. all rules of R with source $f(x_1, \dots, x_n)$, and by $R_{f,a}$ the rules specifying an a -labelled transition for operator f , i.e. all rules of R_f with a conclusion that is a -labelled.

Definition 2.23 (PTSS). A *probabilistic transition system specification* (PTSS) in PGSOS format is a triple $P = (\Sigma, A, R)$, where

- Σ is a signature,
- A is a countable set of actions,
- R is a countable set of PGSOS rules, and
- $R_{f,a}$ is finite for all $f \in \Sigma$ and $a \in A$.

The last property ensures that the supported model (Definition 2.25) is image-finite such that the fixed point characterization of bisimulation metrics coincides with the coinductive characterization (Proposition 2.14).

The operational semantics of terms is given by inductively applying the respective PGSOS rules. Then, a supported model of a PTSS describes the operational semantics of all terms. In other words, a supported model of a PGSOS specification P is a PTS M with transition relation \rightarrow such that \rightarrow contains all and only those transitions for which the rules of P offer a justification.

Definition 2.24 (Supported transition). Let $P = (\Sigma, A, R)$ be a PTSS and $r \in R$ be a rule. Given a PTS $M = (\mathsf{T}(\Sigma), A, \rightarrow)$ and a closed substitution σ , we say that the σ -instance of r is *satisfied* in M and allows to derive $t \xrightarrow{a} \pi$, formally $M \models_r^\sigma t \xrightarrow{a} \pi$, if

- $\sigma(x_i) \xrightarrow{a_{i,k}} \sigma(\mu_{i,k}) \in \rightarrow$ for all $x_i \xrightarrow{a_{i,k}} \mu_{i,k} \in \text{prem}(r)$,
- $\sigma(x_i) \xrightarrow{b_{i,l}} \pi \notin \rightarrow$ for any $\pi \in \Delta(\mathsf{T}(\Sigma))$, for all $x_i \xrightarrow{b_{i,l}} \pi \in \text{prem}(r)$, and
- $t \xrightarrow{a} \pi \in \rightarrow$ for $t \xrightarrow{a} \pi = \sigma(\text{conc}(r))$.

We call a transition $t \xrightarrow{a} \pi$ in M *supported* by P , notation $M \models_P t \xrightarrow{a} \pi$, if there is some $r \in R$ and a closed substitution σ such that $M \models_r^\sigma t \xrightarrow{a} \pi$.

The supported transitions of a PTSS P form the supported model of P .

Definition 2.25 (Supported model). Let $P = (\Sigma, A, R)$ be a PTSS. A PTS $M = (\mathsf{T}(\Sigma), A, \rightarrow)$ is a *supported model* if

$$t \xrightarrow{a} \pi \text{ iff } M \models_P t \xrightarrow{a} \pi$$

for all $t \xrightarrow{a} \pi \in \rightarrow$.

Each PTSS in PGSOS format has a supported model which is moreover unique [BIM95, Bar04]. We call the single supported PTS of a PTSS P also the *induced model* of P .

Intuitively, a term $f(t_1, \dots, t_n)$ represents the composition of terms t_1, \dots, t_n by operator f . A rule r specifies some transition $f(t_1, \dots, t_n) \xrightarrow{a} \pi$ that represents the evolution of the composed term $f(t_1, \dots, t_n)$ by action a to the distribution π .

Definition 2.26 (Disjoint extension [ABV94]). Let $P_1 = (\Sigma_1, A, R_1)$ and $P_2 = (\Sigma_2, A, R_2)$ be two PGSOS PTSSs. P_2 is a *disjoint extension* of P_1 , notation $P_1 \sqsubseteq P_2$, iff $\Sigma_1 \subseteq \Sigma_2$, $R_1 \subseteq R_2$ and R_2 introduces no new rule for any operator in Σ_1 .

3. NON-RECURSIVE PROCESSES

We start by discussing compositional reasoning over probabilistic processes that are composed by non-recursive process combinators. First we introduce the most common non-recursive process combinators, then study the distance between processes composed by these combinators, and conclude by analyzing their compositionality properties. Our study of compositionality properties generalizes earlier results of [DGJP04, DCP06] which considered only a small set of process combinators and only the compositionality property of non-expansiveness. The development of tight bounds on the distance between composed processes (necessary for effective metric assume-guarantee performance validation) is novel.

$\overline{\varepsilon \xrightarrow{\surd} \delta(0)}$	$\overline{a. \bigoplus_{i=1}^n [p_i]x_i \xrightarrow{a} \sum_{i=1}^n p_i \delta(x_i)}$		
$\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x; y \xrightarrow{a} \mu; \delta(y)}$	$\frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{a} \nu}{x; y \xrightarrow{a} \nu}$	$\frac{x \xrightarrow{a} \mu}{x + y \xrightarrow{a} \mu}$	$\frac{y \xrightarrow{a} \nu}{x + y \xrightarrow{a} \nu}$
$\frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu \quad a \neq \surd}{x y \xrightarrow{a} \mu \nu}$		$\frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{\surd} \nu}{x y \xrightarrow{\surd} \delta(0)}$	
$\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x y \xrightarrow{a} \mu \delta(y)}$	$\frac{y \xrightarrow{a} \nu \quad a \neq \surd}{x y \xrightarrow{a} \delta(x) \nu}$	$\frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{\surd} \nu}{x y \xrightarrow{\surd} \delta(0)}$	
$\frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu \quad a \in B \setminus \{\surd\}}{x _B y \xrightarrow{a} \mu _B \nu}$		$\frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{\surd} \nu}{x _B y \xrightarrow{\surd} \delta(0)}$	
$\frac{x \xrightarrow{a} \mu \quad a \notin B \cup \{\surd\}}{x _B y \xrightarrow{a} \mu _B \delta(y)}$		$\frac{y \xrightarrow{a} \nu \quad a \notin B \cup \{\surd\}}{x _B y \xrightarrow{a} \delta(x) _B \nu}$	

Table 1: Standard non-recursive process combinators

3.1. Non-recursive process combinators. We introduce now a probabilistic process algebra that comprises many of the probabilistic process combinators from CCS [JLY01, Bar04, DD07] and CSP [JLY01, Bar04, DvGH⁺07, DL12]. Assume a set of actions A , with $\surd \in A$ denoting the successful termination action. Let Σ_{PA} be the signature with the following operators:

- constants 0 (stop process) and ε (skip process);
- a family of n -ary probabilistic prefix operators $a.([p_1]_-\oplus \dots \oplus [p_n]_-)$ with $a \in A$, $n \geq 1$, $p_1, \dots, p_n \in (0, 1]$ and $\sum_{i=1}^n p_i = 1$;
- binary operators
 - $_-;_-$ (sequential composition),
 - $_-+_-$ (alternative composition),
 - $_-+_p_-$ (probabilistic alternative composition), with $p \in (0, 1)$,
 - $_-|_-$ (synchronous parallel composition),
 - $_-||_-$ (asynchronous parallel composition),
 - $_-|||_p_-$ (probabilistic parallel composition), with $p \in (0, 1)$, and
 - $_-||_B_-$ for each for each $B \subseteq A$ (CSP-like parallel composition).

The PTSS $P_{\text{PA}} = (\Sigma_{\text{PA}}, A, R_{\text{PA}})$ is given by the set of PGSOS rules R_{PA} in Table 1 and Table 2.

The probabilistic prefix operator expresses that the process $a.([p_1]t_1 \oplus \dots \oplus [p_n]t_n)$ can perform action a and evolves to process t_i with probability p_i . Sometimes we write $a. \bigoplus_{i=1}^n [p_i]t_i$ for $a.([p_1]t_1 \oplus \dots \oplus [p_n]t_n)$ and $a.t$ for $a.([1]t)$ (deterministic prefix operator). The sequential composition and the alternative composition are as usual. The synchronous parallel composition $t | t'$ describes the simultaneous evolution of processes t and t' , while the asynchronous parallel composition $t || t'$ describes the interleaving of t and t' where both processes can progress by alternating

$$\begin{array}{c}
\frac{x \xrightarrow{a} \mu \quad y \not\xrightarrow{a}}{x +_p y \xrightarrow{a} \mu} \quad \frac{x \not\xrightarrow{a} \quad y \xrightarrow{a} \nu}{x +_p y \xrightarrow{a} \nu} \quad \frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu}{x +_p y \xrightarrow{a} \mu \oplus_p \nu} \\
\\
\frac{x \xrightarrow{a} \mu \quad y \not\xrightarrow{a} \quad a \neq \surd}{x \parallel_p y \xrightarrow{a} \mu \parallel_p \delta(y)} \quad \frac{x \not\xrightarrow{a} \quad y \xrightarrow{a} \nu \quad a \neq \surd}{x \parallel_p y \xrightarrow{a} \delta(x) \parallel_p \nu} \\
\\
\frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu \quad a \neq \surd}{x \parallel_p y \xrightarrow{a} \mu \parallel_p \delta(y) \oplus_p \delta(x) \parallel_p \nu} \quad \frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{\surd} \nu}{x \parallel_p y \xrightarrow{\surd} \delta(0)}
\end{array}$$

Table 2: Standard non-recursive probabilistic process combinators

at any rate the execution of their actions. The CSP-like parallel composition $t \parallel_B t'$ describes multi-party synchronization where t and t' synchronize on actions in B and evolve independently for all other actions.

The probabilistic variants of the alternative composition and the asynchronous parallel composition replace the nondeterministic choice of their non-probabilistic variant by a probabilistic choice. The probabilistic alternative composition $t +_p t'$ evolves to the probabilistic choice between a distribution reached by t (with probability p) and a distribution reached by t' (with probability $1 - p$) for actions which can be performed by both processes. For actions that can be performed by either only t or only t' , the probabilistic alternative composition $t +_p t'$ behaves just like the nondeterministic alternative composition $t + t'$. Similarly, the probabilistic parallel composition $t \parallel_p t'$ evolves to a probabilistic choice (with respectively the probability p and $1 - p$) between the two nondeterministic choices of the nondeterministic parallel composition $t \parallel t'$ for actions which can be performed by both t and t' . For actions that can be performed by either only t or only t' , the probabilistic parallel composition $t \parallel_p t'$ behaves just like the nondeterministic parallel composition $t \parallel t'$.

3.2. Distance between processes combined by non-recursive process combinators. We develop now tight bounds on the distance between processes combined by the non-recursive process combinators presented in Table 1 and Table 2. This will allow us to derive the compositionality properties of those operators. As we will discuss two different compositionality properties for non-recursive process combinators (non-extensiveness, Definition 3.4, and non-expansiveness, Definition 3.7), we split in this section the discussion on the distance bounds accordingly. We use disjoint extensions of the specification of the process combinators in order to reason over the composition of arbitrary processes.

We will express the bound on the distance between composed processes $f(s_1, \dots, s_n)$ and $f(t_1, \dots, t_n)$ in terms of the distance between their respective components s_i and t_i . Intuitively, given a probabilistic process $f(s_1, \dots, s_n)$ we provide a bound on the distance to the respective probabilistic process $f(t_1, \dots, t_n)$ where each component s_i is replaced by the component t_i .

We start with those process combinators that satisfy the later discussed compositionality property of non-extensiveness (Definition 3.4).

Proposition 3.1. *Let $P = (\Sigma, A, R)$ be any PTSS with $P_{PA} \sqsubseteq P$. For all terms $s_i, t_i \in \mathcal{T}(\Sigma)$ it holds:*

- (a) $\mathbf{d}(a. \bigoplus_{i=1}^n [p_i]s_i, a. \bigoplus_{i=1}^n [p_i]t_i) \leq \lambda \cdot \sum_{i=1}^n p_i \mathbf{d}(s_i, t_i)$;
- (b) $\mathbf{d}(s_1 + s_2, t_1 + t_2) \leq \max(\mathbf{d}(s_1, t_1), \mathbf{d}(s_2, t_2))$;

(c) $\mathbf{d}(s_1 +_p s_2, t_1 +_p t_2) \leq \max(\mathbf{d}(s_1, t_1), \mathbf{d}(s_2, t_2))$.

Proof. First we consider the probabilistic prefix operator (Proposition 3.1.(a)). The only transitions from $a. \bigoplus_{i=1}^n [p_i]s_i$ and $a. \bigoplus_{i=1}^n [p_i]t_i$ are $a. \bigoplus_{i=1}^n [p_i]s_i \xrightarrow{a} \sum_{i=1}^n p_i \delta(s_i)$ and $a. \bigoplus_{i=1}^n [p_i]t_i \xrightarrow{a} \sum_{i=1}^n p_i \delta(t_i)$. Hence we need to show that $\lambda \cdot \mathbf{K}(\mathbf{d})(\sum_{i=1}^n p_i \delta(s_i), \sum_{i=1}^n p_i \delta(t_i)) \leq \lambda \cdot \sum_{i=1}^n p_i \mathbf{d}(s_i, t_i)$. This property can be derived by Proposition 2.19 as follows:

$$\begin{aligned} & \mathbf{K}(\mathbf{d})\left(\sum_{i=1}^n p_i \delta(s_i), \sum_{i=1}^n p_i \delta(t_i)\right) \\ & \leq \sum_{i=1}^n p_i \mathbf{K}(\mathbf{d})(\delta(s_i), \delta(t_i)) && \text{(Proposition 2.19.3)} \\ & = \sum_{i=1}^n p_i \mathbf{d}(s_i, t_i) && \text{(Proposition 2.19.2)} \end{aligned}$$

We proceed with the alternative composition operator (Proposition 3.1.(b)). If either $\mathbf{d}(s_1, t_1) = 1$ or $\mathbf{d}(s_2, t_2) = 1$ then the statement is trivial since \mathbf{d} is a 1-bounded pseudometric. Hence, we assume $\mathbf{d}(s_1, t_1) < 1$ and $\mathbf{d}(s_2, t_2) < 1$. We consider now the two different rules specifying the alternative composition operator and show that in each case whenever $s_1 + s_2 \xrightarrow{a} \pi$ is derivable by some of the rules then there is a transition $t_1 + t_2 \xrightarrow{a} \pi'$ derivable by the same rule s.t. $\lambda \cdot \mathbf{K}(\mathbf{d})(\pi, \pi') \leq \max(\mathbf{d}(s_1, t_1), \mathbf{d}(s_2, t_2))$.

- (1) Assume that $s_1 + s_2 \xrightarrow{a} \pi$ is derived from $s_1 \xrightarrow{a} \pi$. Since $\mathbf{d}(s_1, t_1) < 1$ and \mathbf{d} satisfies the transfer condition of the bisimulation metrics, there exists a transition $t_1 \xrightarrow{a} \pi'$ for a distribution π' with $\lambda \cdot \mathbf{K}(\mathbf{d})(\pi, \pi') \leq \mathbf{d}(s_1, t_1) \leq \max(\mathbf{d}(s_1, t_1), \mathbf{d}(s_2, t_2))$. Finally, from $t_1 \xrightarrow{a} \pi'$ we derive $t_1 + t_2 \xrightarrow{a} \pi'$.
- (2) Assume that $s_1 + s_2 \xrightarrow{a} \pi$ is derived from $s_2 \xrightarrow{a} \pi$. The argument is the same of the previous case.

We conclude with the probabilistic alternative composition operator (Proposition 3.1.(c)). If either $\mathbf{d}(s_1, t_1) = 1$ or $\mathbf{d}(s_2, t_2) = 1$ then the statement is trivial since \mathbf{d} is a 1-bounded pseudometric. Hence, we assume $\mathbf{d}(s_1, t_1) < 1$ and $\mathbf{d}(s_2, t_2) < 1$. We consider now the three different rules specifying the probabilistic alternative composition operator and show that in each case whenever $s_1 + s_2 \xrightarrow{a} \pi$ is derivable by some of the rules then there is a transition $t_1 + t_2 \xrightarrow{a} \pi'$ derivable by the same rule s.t. $\lambda \cdot \mathbf{K}(\mathbf{d})(\pi, \pi') \leq \max(\mathbf{d}(s_1, t_1), \mathbf{d}(s_2, t_2))$.

- (1) Assume that $s_1 +_p s_2 \xrightarrow{a} \pi$ is derived from $s_1 \xrightarrow{a} \pi$ and $s_2 \not\xrightarrow{a}$. Since $\mathbf{d}(s_1, t_1) < 1$ and \mathbf{d} satisfies the transfer condition of the bisimulation metrics, there exists a transition $t_1 \xrightarrow{a} \pi'$ with $\lambda \cdot \mathbf{K}(\mathbf{d})(\pi, \pi') \leq \mathbf{d}(s_1, t_1) \leq \max(\mathbf{d}(s_1, t_1), \mathbf{d}(s_2, t_2))$. Since $\mathbf{d}(s_2, t_2) < 1$, by Proposition 2.17.2 the processes s_2 and t_2 agree on the actions they can perform immediately. Thus $t_2 \not\xrightarrow{a}$. Hence we can derive the transition $t_1 +_p t_2 \xrightarrow{a} \pi'$.
- (2) Assume that $s_1 +_p s_2 \xrightarrow{a} \pi$ is derived from $s_1 \not\xrightarrow{a}$ and $s_2 \xrightarrow{a} \pi$. The argument is the same of the previous case.
- (3) Assume that $s_1 +_p s_2 \xrightarrow{a} \pi$ with $\pi = p(\pi_1) + (1-p)\pi_2$ is derived from $s_1 \xrightarrow{a} \pi_1$ and $s_2 \xrightarrow{a} \pi_2$. Then, since $\mathbf{d}(s_1, t_1) < 1$ and $\mathbf{d}(s_2, t_2) < 1$ and \mathbf{d} satisfies the transfer condition of the bisimulation metrics, there exist transitions $t_1 \xrightarrow{a} \pi'_1$ with $\lambda \cdot \mathbf{K}(\mathbf{d})(\pi_1, \pi'_1) \leq \mathbf{d}(s_1, t_1)$ and $t_2 \xrightarrow{a}$

π'_2 with $\lambda \cdot \mathbf{K}(\mathbf{d})(\pi_2, \pi'_2) \leq \mathbf{d}(s_2, t_2)$. Therefore we derive $t_1 +_p t_2 \xrightarrow{a} p\pi'_1 + (1-p)\pi'_2$, with

$$\begin{aligned} & \lambda \cdot \mathbf{K}(\mathbf{d})(p\pi_1 + (1-p)\pi_2, p\pi'_1 + (1-p)\pi'_2) \\ & \leq \lambda \cdot (p \mathbf{K}(\mathbf{d})(\pi_1, \pi'_1) + (1-p) \mathbf{K}(\mathbf{d})(\pi_2, \pi'_2)) && \text{(Proposition 2.19.3)} \\ & \leq \lambda \cdot \max(\mathbf{K}(\mathbf{d})(\pi_1, \pi'_1), \mathbf{K}(\mathbf{d})(\pi_2, \pi'_2)) \\ & \leq \max(\mathbf{d}(s_1, t_1), \mathbf{d}(s_2, t_2)). \end{aligned} \quad \square$$

We note that the distance between action prefixed processes (Proposition 3.1.(a)) is discounted by λ since the processes $a. \bigoplus_{i=1}^n [p_i]s_i$ and $a. \bigoplus_{i=1}^n [p_i]t_i$ perform first the action a before the processes s_i and t_i may evolve and their distance is observed. The distances between processes composed by either the nondeterministic alternative composition operator or by the probabilistic alternative composition operator are both bounded by the maximum of the distances between their respective arguments (Propositions 3.1.(b) and 3.1.(c)). The distance bounds for these operators coincide since the first two rules specifying the probabilistic alternative composition define the same operational behavior as the nondeterministic alternative composition and the third rule defining a convex combination of these transitions applies only for those actions that can be performed by both processes s_1 and s_2 and resp. t_1 and t_2 . If the probabilistic alternative composition would be defined by only the third rule of Table 2, then $\mathbf{d}(s_1 +_p s_2, t_1 +_p t_2) \leq p\mathbf{d}(s_1, t_1) + (1-p)\mathbf{d}(s_2, t_2)$.

Finally, we note that the processes s_i and t_i in Propositions 3.1 are obtained by using arbitrary operators in Σ (not necessarily only operators in Σ_{PA}).

We proceed with those process combinators that satisfy the later discussed compositionality property of non-expansiveness (Definition 3.7).

Proposition 3.2. *Let $P = (\Sigma, A, R)$ be any PTSS with $P_{PA} \sqsubseteq P$. For all terms $s_i, t_i \in T(\Sigma)$ it holds:*

$$(a) \quad \mathbf{d}(s_1; s_2, t_1; t_2) \leq \begin{cases} 1 & \text{if } \mathbf{d}(s_1, t_1) = 1 \\ \max(d_{1,2}^1, \mathbf{d}(s_2, t_2)) & \text{if } \mathbf{d}(s_1, t_1) \in [0, 1) \end{cases}$$

$$(b) \quad \mathbf{d}(s_1 \mid s_2, t_1 \mid t_2) \leq d^s$$

$$(c) \quad \mathbf{d}(s_1 \parallel s_2, t_1 \parallel t_2) \leq d^a$$

$$(d) \quad \mathbf{d}(s_1 \parallel_B s_2, t_1 \parallel_B t_2) \leq \begin{cases} d^s & \text{if } B \setminus \{\surd\} \neq \emptyset \\ d^a & \text{otherwise} \end{cases}$$

$$(e) \quad \mathbf{d}(s_1 \parallel_p s_2, t_1 \parallel_p t_2) \leq d^a$$

with

$$d^s = \begin{cases} 1 & \text{if } \mathbf{d}(s_1, t_1) = 1 \\ 1 & \text{if } \mathbf{d}(s_2, t_2) = 1 \\ d_{1,2}^0 & \text{otherwise} \end{cases}$$

$$d^a = \begin{cases} 1 & \text{if } \mathbf{d}(s_1, t_1) = 1 \\ 1 & \text{if } \mathbf{d}(s_2, t_2) = 1 \\ \max(d_{1,2}^2, d_{2,1}^2) & \text{otherwise} \end{cases}$$

$$d_{1,2}^n = \mathbf{d}(s_1, t_1) + \lambda^n(1 - \mathbf{d}(s_1, t_1)/\lambda)\mathbf{d}(s_2, t_2)$$

$$d_{2,1}^n = \mathbf{d}(s_2, t_2) + \lambda^n(1 - \mathbf{d}(s_2, t_2)/\lambda)\mathbf{d}(s_1, t_1)$$

Proof. We will prove only Proposition 3.2.(d) (CSP-like parallel composition \parallel_B). The synchronous and asynchronous parallel composition operators (Propositions 3.2.(b) and 3.2.(c)) are special cases, since $|$ coincides with \parallel_A and \parallel coincides with \parallel_\emptyset . The proofs for the probabilistic parallel composition operator \parallel_p (Proposition 3.2.(e)) and the sequential composition $;$ (Proposition 3.2.(a)) are analogous.

We prove the case $B \setminus \{\surd\} \neq \emptyset$ (the case $B \setminus \{\surd\} = \emptyset$ is similar). First we need to introduce the notion of congruence closure for λ -bisimilarity metric \mathbf{d} as the quantitative analogue of the well-known concept of congruence closure of a process equivalence. We define the metric congruence closure of \mathbf{d} for operator \parallel_B w.r.t. the bound provided in Proposition 3.2.(d) as a function $d: \mathbb{T}(\Sigma) \times \mathbb{T}(\Sigma) \rightarrow [0, 1]$ defined by

$$d(t, t') = \begin{cases} \min(\lambda[1 - (1 - d(t_1, t'_1))/\lambda](1 - d(t_2, t'_2)/\lambda), \mathbf{d}(t, t')) & \text{if } \begin{cases} t = t_1 \parallel_B t_2 \wedge \\ t' = t'_1 \parallel_B t'_2 \wedge \\ \mathbf{d}(t_1, t'_1) < 1 \wedge \\ \mathbf{d}(t_2, t'_2) < 1 \end{cases} \\ \mathbf{d}(t, t') & \text{otherwise} \end{cases}$$

We note that d satisfies by construction $d(s_1 \parallel_B s_2, t_1 \parallel_B t_2) \leq d^s$ since $\lambda[1 - (1 - d(s_1, t_1))/\lambda](1 - d(s_2, t_2)/\lambda) = d(s_1, t_1) + (1 - d(s_1, t_1)/\lambda)d(s_2, t_2)$. We note also that d satisfies by construction $d \sqsubseteq \mathbf{d}$. It remains to show that $\mathbf{d} \sqsubseteq d$, thus giving $\mathbf{d} = d$, and Proposition 3.2.(d) holds. Since \mathbf{d} is the least prefixed point of \mathbf{B} , to show $\mathbf{d} \sqsubseteq d$ it is enough to prove that d is a prefixed point of \mathbf{B} .

To prove that $\mathbf{B}(d) \sqsubseteq d$ we need to show that d satisfies the transfer condition of the bisimulation metrics, namely

$$\text{for all } t \xrightarrow{a} \pi \text{ there exists a transition } t' \xrightarrow{a} \pi' \text{ with } \lambda \cdot \mathbf{K}(d)(\pi, \pi') \leq d(t, t') \quad (3.1)$$

for all terms $t, t' \in \mathbb{T}(\Sigma)$ with $d(t, t') < 1$.

We prove Equation 3.1 by induction over the overall number k of occurrences of operator \parallel_B occurring in t and t' .

Consider the base case $k = 0$. By definition of d , we have that $d(t, t') = \mathbf{d}(t, t')$. Since $\mathbf{d}(t, t') < 1$ we are sure that the transition $t \xrightarrow{a} \pi$ is mimicked by some transition $t' \xrightarrow{a} \pi'$ for some distribution $\pi' \in \Delta(\mathbb{T}(\Sigma))$ such that $\lambda \cdot \mathbf{K}(\mathbf{d})(\pi, \pi') \leq \mathbf{d}(t, t')$. By Proposition 2.19.1 from $d \sqsubseteq \mathbf{d}$ we infer $\mathbf{K}(d) \sqsubseteq \mathbf{K}(\mathbf{d})$. Therefore we conclude

$$\lambda \cdot \mathbf{K}(d)(\pi, \pi') \leq \lambda \cdot \mathbf{K}(\mathbf{d})(\pi, \pi') \leq \mathbf{d}(t, t') = d(t, t')$$

which confirms that Equation 3.1 holds for t and t' .

Consider the inductive step $k > 0$. If either t is not of the form $t = t_1 \parallel_B t_2$, or t' is not of the form $t' = t'_1 \parallel_B t'_2$, then by definition of d we have $d(t, t') = \mathbf{d}(t, t')$ and Equation 3.1 follows precisely as in the base case $k = 0$. If both $t = t_1 \parallel_B t_2$ and $t' = t'_1 \parallel_B t'_2$, then we distinguish two cases, namely $d(t, t') = \mathbf{d}(t, t')$ (either $\mathbf{d}(t_1, t'_1) = 1$ or $\mathbf{d}(t_2, t'_2) = 1$ or $\mathbf{d}(t, t') < \lambda[1 - (1 - d(t_1, t'_1))/\lambda](1 - d(t_2, t'_2)/\lambda)$) and $d(t, t') = \lambda[1 - (1 - d(t_1, t'_1))/\lambda](1 - d(t_2, t'_2)/\lambda)$ (both $\mathbf{d}(t_1, t'_1) < 1$ and $\mathbf{d}(t_2, t'_2) < 1$ and $\mathbf{d}(t, t') \geq \lambda[1 - (1 - d(t_1, t'_1))/\lambda](1 - d(t_2, t'_2)/\lambda)$). In case $d(t, t') = \mathbf{d}(t, t')$ Equation 3.1 follows precisely as in the base case $k = 0$. Consider the case $d(t, t') = \lambda[1 - (1 - d(t_1, t'_1))/\lambda](1 - d(t_2, t'_2)/\lambda)$. We have four different subcases:

- (1) $t_1 \xrightarrow{a} \pi_1, t_2 \xrightarrow{a} \pi_2, a \in B \setminus \{\surd\}$ and $\pi = \pi_1 \parallel_B \pi_2$;
- (2) $t_1 \xrightarrow{a} \pi_1, t_2 \xrightarrow{a} \delta, a \notin B \cup \{\surd\}$ and $\pi = \pi_1 \parallel_B \delta(t_2)$;
- (3) $t_2 \xrightarrow{a} \pi_2, t_1 \xrightarrow{a} \delta, a \notin B \cup \{\surd\}$ and $\pi = \delta(t_1) \parallel_B \pi_2$;
- (4) $t_1 \xrightarrow{a} \pi_1, t_2 \xrightarrow{a} \pi_2, a = \surd$ and $\pi = \delta(0)$.

We start with the first case. By $\mathbf{d}(t_1, t'_1) < 1$ and $\mathbf{d}(t_2, t'_2) < 1$ and $d \sqsubseteq \mathbf{d}$, we get $d(t_1, t'_1) < 1$ and $d(t_2, t'_2) < 1$. By the inductive hypothesis we get that there are also transitions $t'_1 \xrightarrow{a} \pi'_1$ and $t'_2 \xrightarrow{a} \pi'_2$ with $\lambda \cdot \mathbf{K}(d)(\pi_1, \pi'_1) \leq d(t_1, t'_1)$ and $\lambda \cdot \mathbf{K}(d)(\pi_2, \pi'_2) \leq d(t_2, t'_2)$. Hence, there is also the transition $t'_1 \parallel_B t'_2 \xrightarrow{a} \pi'_1 \parallel_B \pi'_2$. Then

$$\begin{aligned} & \lambda \cdot \mathbf{K}(d)(\pi_1 \parallel_B \pi_2, \pi'_1 \parallel_B \pi'_2) \\ & \leq \lambda^2 [1 - (1 - \mathbf{K}(d)(\pi_1, \pi'_1)/\lambda)(1 - \mathbf{K}(d)(\pi_2, \pi'_2)/\lambda)] \\ & \leq \lambda^2 [1 - (1 - d(t_1, t'_1)/\lambda^2)(1 - d(t_2, t'_2)/\lambda^2)] \\ & \leq \lambda [1 - (1 - d(t_1, t'_1)/\lambda)(1 - d(t_2, t'_2)/\lambda)] \\ & = d(t_1 \parallel_B t_2, t'_1 \parallel_B t'_2) \end{aligned}$$

with the first step by Theorem 2.20 (using the fact that the candidate modulus of continuity of operator \parallel_B given by $z(\epsilon_1, \epsilon_2) = \lambda[1 - (1 - \epsilon_1/\lambda)(1 - \epsilon_2/\lambda)]$ is concave) and the second step by the inductive hypothesis $\lambda \cdot \mathbf{K}(d)(\pi_i, \pi'_i) \leq d(t_i, t'_i)$. Thus, the metric bisimulation transfer condition (Equation 3.1) is satisfied for d in this case.

Consider now the second case. By $\mathbf{d}(t_1, t'_1) < 1$ and $d \sqsubseteq \mathbf{d}$, we get $d(t_1, t'_1) < 1$. By the inductive hypothesis we get that there is also a transitions $t'_1 \xrightarrow{a} \pi'_1$ with $\lambda \cdot \mathbf{K}(d)(\pi_1, \pi'_1) \leq d(t_1, t'_1)$. By Proposition 2.17.2 we have that $t'_2 \not\xrightarrow{a}$, therefore we can derive the transition $t'_1 \parallel_B t'_2 \xrightarrow{a} \pi'_1 \parallel_B \delta(t'_2)$. Then

$$\begin{aligned} & \lambda \cdot \mathbf{K}(d)(\pi_1 \parallel_B \delta(t_2), \pi'_1 \parallel_B \delta(t'_2)) \\ & \leq \lambda^2 [1 - (1 - \mathbf{K}(d)(\pi_1, \pi'_1)/\lambda)(1 - \mathbf{K}(d)(\delta(t_2), \delta(t'_2))/\lambda)] \\ & \leq \lambda^2 [1 - (1 - d(t_1, t'_1)/\lambda^2)(1 - d(t_2, t'_2)/\lambda)] \\ & \leq \lambda [1 - (1 - d(t_1, t'_1)/\lambda)(1 - d(t_2, t'_2)/\lambda)] \\ & = d(t_1 \parallel_B t_2, t'_1 \parallel_B t'_2) \end{aligned}$$

with step 1 again from Theorem 2.20 like in the first case and the second step by the inductive hypothesis $\lambda \cdot \mathbf{K}(d)(\pi_1, \pi'_1) \leq d(t_1, t'_1)$ and Proposition 2.19.2. Hence, the metric bisimulation transfer condition (Equation 3.1) is satisfied for d in this case.

The third case is analogous to the second one.

Consider now the fourth case. By $\mathbf{d}(t_1, t'_1) < 1$ and $\mathbf{d}(t_2, t'_2) < 1$ and $d \sqsubseteq \mathbf{d}$, we get $d(t_1, t'_1) < 1$ and $d(t_2, t'_2) < 1$. By the inductive hypothesis we get that there are also transitions $t'_1 \xrightarrow{\vee} \pi'_1$ and $t'_2 \xrightarrow{\vee} \pi'_2$. Hence, there is also the transition $t'_1 \parallel_B t'_2 \xrightarrow{\vee} \delta(0)$. Then $\lambda \cdot \mathbf{K}(d)(\delta(0), \delta(0)) = 0 \leq d(t_1 \parallel_B t_2, t'_1 \parallel_B t'_2)$. Thus, the metric bisimulation transfer condition (Equation 3.1) is satisfied for d also in this case. \square

The expression d^s in Proposition 3.2 captures the distance bound between the synchronously evolving processes s_1 and s_2 on the one hand and the synchronously evolving processes t_1 and t_2 on the other hand. We remark that the distances $\mathbf{d}(s_1, t_1)$ and $\mathbf{d}(s_2, t_2)$ contribute symmetrically to d^s since $d_{1,2}^0 = \mathbf{d}(s_1, t_1) + (1 - \mathbf{d}(s_1, t_1)/\lambda)\mathbf{d}(s_2, t_2) = \mathbf{d}(s_2, t_2) + (1 - \mathbf{d}(s_2, t_2)/\lambda)\mathbf{d}(s_1, t_1) = d_{2,1}^0$. The expressions $d_{1,2}^n, d_{2,1}^n$ with $n > 0$ cover different scenarios of the asynchronous evolution of those processes. The expression $d_{1,2}^n$ (resp. $d_{2,1}^n$) denotes the distance bound between the asynchronously evolving processes s_1 and s_2 on the one hand and the asynchronously evolving processes t_1 and t_2 on the other hand, at which the first n transitions are performed by the processes s_1 and t_1 (resp. the first n transitions are performed by processes s_2 and t_2).

If $\mathbf{d}(s_1, t_1) = 1$ or $\mathbf{d}(s_2, t_2) = 1$, then the processes s_1 and t_1 and the processes s_2 and t_2 may disagree on the initial actions they can perform, and also the composed processes may disagree on their initial actions and have then also the maximal distance of 1 (cf. Proposition 2.17 and Remark 2.18). We analyze the bound for the process combinators in details assuming both $\mathbf{d}(s_1, t_1) < 1$ and $\mathbf{d}(s_2, t_2) < 1$.

The distance between the sequentially composed processes $s_1; s_2$ and $t_1; t_2$ (Proposition 3.2.(a)) is given if $\mathbf{d}(s_1, t_1) \in [0, 1)$ as the maximum of

- (i) distance $d_{1,2}^1 = \mathbf{d}(s_1, t_1) + \lambda(1 - \mathbf{d}(s_1, t_1)/\lambda)\mathbf{d}(s_2, t_2)$, which captures the case that first the processes s_1 and t_1 evolve followed by s_2 and t_2 , and
- (ii) distance $\mathbf{d}(s_2, t_2)$, which captures the case that the processes s_2 and t_2 evolve immediately because both s_1 and t_1 terminate successfully at their first computation step.

The distance $d_{1,2}^1$ weights the distance $\mathbf{d}(s_2, t_2)$ between s_2 and t_2 by $\lambda(1 - \mathbf{d}(s_1, t_1)/\lambda)$. The discount λ expresses that processes s_2 and t_2 are delayed by at least one transition step whenever s_1 and t_1 perform at least one transition step before terminating. Additionally, note that the difference between s_2 and t_2 can only be observed when s_1 and t_1 agree to terminate. When processes s_1 and t_1 evolve by one step, they disagree by $\mathbf{d}(s_1, t_1)/\lambda$ on their behavior. Hence they agree by $(1 - \mathbf{d}(s_1, t_1)/\lambda)$. Thus, the distance between processes s_2 and t_2 needs to be additionally weighted by $(1 - \mathbf{d}(s_1, t_1)/\lambda)$. In case ((ii)) the distance between s_2 and t_2 is not discounted since both processes start immediately.

The distance bound between synchronous parallel composed processes $s_1 \mid s_2$ and $t_1 \mid t_2$ (Proposition 3.2.(b)) is the expression d^s , which is $d_{1,2}^0 = \mathbf{d}(s_1, t_1) + (1 - \mathbf{d}(s_1, t_1)/\lambda)\mathbf{d}(s_2, t_2) = \mathbf{d}(s_2, t_2) + (1 - \mathbf{d}(s_2, t_2)/\lambda)\mathbf{d}(s_1, t_1) = d_{2,1}^0$, when both $\mathbf{d}(s_1, t_1) < 1$ and $\mathbf{d}(s_2, t_2) < 1$. Hence the distance between $s_1 \mid s_2$ and $t_1 \mid t_2$ is bounded by the sum of the distance between s_1 and t_1 , which is the degree of dissimilarity between s_1 and t_1 , and the distance between s_2 and t_2 weighted by the probability that s_1 and t_1 agree on their behavior, which is the degree of dissimilarity between s_2 and t_2 under equal behavior of s_1 and t_1 . Alternatively, by $d_{1,2}^0 = d_{2,1}^0 = \lambda(1 - (1 - \mathbf{d}(s_1, t_1)/\lambda)(1 - \mathbf{d}(s_2, t_2)/\lambda))$, the bound to the distance between $s_1 \mid s_2$ and $t_1 \mid t_2$ can be understood as composing processes on the behavior they agree upon, i.e. $s_1 \mid s_2$ and $t_1 \mid t_2$ agree on their behavior if s_1 and t_1 agree (probability of similarity $1 - \mathbf{d}(s_1, t_1)/\lambda$) and if s_2 and t_2 agree (probability of similarity $1 - \mathbf{d}(s_2, t_2)/\lambda$). The resulting distance is then the probability of dissimilarity of the respective behavior $1 - (1 - \mathbf{d}(s_1, t_1)/\lambda)(1 - \mathbf{d}(s_2, t_2)/\lambda)$ multiplied by the discount factor λ .

The distance bound between asynchronous parallel composed processes $s_1 \parallel s_2$ and $t_1 \parallel t_2$ is the expression d^a (Proposition 3.2.(c)). Hence the distance bound is the maximum of $d_{1,2}^2$, namely the distance observable when first processes s_1 and t_1 evolve by at least two transition steps and then s_2 and t_2 , and $d_{2,1}^2$, namely the distance observable when first processes s_2 and t_2 evolve by at least two transition steps and then s_1 and t_1 . Notice that at least two transition steps by the faster processes are necessary to observe their distance before the slower processes start. The behaviors where either s_1 and t_1 perform the first transition step and s_2 and t_2 perform the second transition step, or s_2 and t_2 perform the first transition step and s_1 and t_1 perform the second transition step, give rise to a lower distance wrt. that expressed by the maximum between $d_{1,2}^2$ and $d_{2,1}^2$. The reason is that the observation of the different behaviors is delayed by more transition steps and, therefore, more discounted. Notice that both $d_{1,2}^2$ and $d_{2,1}^2$ differ from the distance d^s of the synchronously evolving processes $s_1 \mid s_2$ and $t_1 \mid t_2$ only by the discount factor λ^2 that is applied to the distance of the delayed processes. Moreover, $d_{1,2}^2$ differs from the distance $d_{1,2}^1$ of the sequential composed processes $s_1; s_2$ and $t_1; t_2$ by the different discount factor that is applied to the distance of the processes s_2 and t_2 . The discount factor in case $d_{1,2}^2$ is λ^2 since s_2 and t_2 are delayed by at least two transition steps after the

distance between s_1 and t_1 is observed, whereas the discount factor in case $d_{1,2}^1$ is λ since the distance between s_1 and t_1 observed at their second transition step may be realized by the ability/inability of performing action \surd , which let s_2 and t_2 start immediately (namely already in this second transition step).

Processes that are composed by the CSP-like parallel composition operator $- \parallel_B -$ evolve synchronously for actions in $B \setminus \{\surd\}$, evolve asynchronously for actions in $A \setminus (B \cup \{\surd\})$, and the action \surd leads always to the stop process if both processes can perform \surd . Since $d^s \geq d^a$, the distance between processes $s_1 \parallel s_2$ and $t_1 \parallel t_2$ (Proposition 3.2.(d)) is bounded by d^s if there is at least one action $a \in B$ with $a \neq \surd$ for which the composed processes can evolve synchronously, and otherwise by d^a .

The distance between processes composed by the probabilistic parallel composition operator $s_1 \parallel_p s_2$ and $t_1 \parallel_p t_2$ (Proposition 3.2.(e)) is bounded by the expression d^a since the first two rules specifying the probabilistic parallel composition define the same operational behavior as the nondeterministic parallel composition, and the third rule defining a convex combination of these transitions applies only for those actions that can be performed by both processes s_1 and s_2 and resp. t_1 and t_2 .

The distance bounds on the distance between processes composed by non-recursive process combinators (Proposition 3.1 and 3.2) are tight.

Proposition 3.3. *Let $\epsilon_i \in [0, 1]$. There are processes $s_i, t_i \in \mathcal{T}(\Sigma_{PA})$ with $\mathbf{d}(s_i, t_i) = \epsilon_i$ such that the inequalities in Propositions 3.1 and 3.2 become equalities.*

Proof. We start with Proposition 3.1. Let $A = \{a_1, \dots, a_n\} \cup \{\surd\}$. We define now the witness processes

- $s_i = t_i = a_i.\varepsilon$, if $\epsilon_i = 0$;
- $s_i = a_i.([1 - \epsilon_i/\lambda]\varepsilon \oplus [\epsilon_i/\lambda]0)$ and $t_i = a_i.\varepsilon$, if $\epsilon_i \in (0, \lambda)$;
- $s_i = a_i.0$ and $t_i = a_i.\varepsilon$, if $\epsilon_i = \lambda < 1$;
- $s_i = 0$ and $t_i = a_i.\varepsilon$, if $\epsilon_i = 1$.

It is easy to see that these processes yield for all process combinators of Proposition 3.1 exactly the stated upper bound.

We proceed now with Propositions 3.2.(a), 3.2.(b) and 3.2.(d), case $B \setminus \{\surd\} \neq \emptyset$. Let $A = \{a, \surd\}$ with $a \in B$. We define now the witness processes

- $s_i = t_i = a.\varepsilon$, if $\epsilon_i = 0$;
- $s_i = a.([1 - \epsilon_i/\lambda]\varepsilon \oplus [\epsilon_i/\lambda]0)$ and $t_i = a.\varepsilon$, if $\epsilon_i \in (0, \lambda)$;
- $s_i = a.0$ and $t_i = a.\varepsilon$, if $\epsilon_i = \lambda < 1$;
- $s_i = 0$ and $t_i = a.\varepsilon$ if $\epsilon_i = 1$.

These processes yield for all process combinators of Propositions 3.2.(a), 3.2.(b) and 3.2.(d), case $B \setminus \{\surd\} \neq \emptyset$, exactly the stated upper bound.

Finally, we conclude with Propositions 3.2.(c), 3.2.(e) and 3.2.(d), case $B \setminus \{\surd\} = \emptyset$. Let $A = \{a_1, a_2, a\} \cup \{\surd\}$. We define now the witness processes

- $s_i = t_i = a_i.a.0$, if $\epsilon_i = 0$;
- $s_i = a_i.([1 - \epsilon_i/\lambda]a.0 \oplus [\epsilon_i/\lambda]0)$ and $t_i = a_i.a.0$, if $\epsilon_i \in (0, \lambda)$;
- $s_i = a_i.0$ and $t_i = a_i.a.0$, if $\epsilon_i = \lambda < 1$;
- $s_i = 0$ and $t_i = a_i.\varepsilon$, if $\epsilon_i = 1$.

These processes yield for all process combinators of Propositions 3.2.(c), 3.2.(e) and 3.2.(d), case $B \setminus \{\surd\} = \emptyset$, exactly the stated upper bound. \square

3.3. Compositional reasoning over non-recursive processes. In order to specify and verify systems in a compositional manner, it is necessary that the behavioral semantics is compatible with all operators of the language that describe these systems. There are multiple proposals which properties of process combinators facilitate compositional reasoning. In this section we discuss non-extensiveness [BBLM13] and non-expansiveness [DJGP02, DGJP04, DCP06, CGPX14]), which are compositionality properties based on the p -norm. They allow for compositional reasoning over probabilistic processes that are built of non-recursive process combinators. Non-extensiveness and non-expansiveness are very strong forms of uniform continuity. For instance, a non-expansive operator ensures that the distance between the composed processes is at most the sum of the distances between their parts. Later in Section 4.3 we will propose uniform continuity as generalization of these properties that allows also for compositional reasoning over recursive processes.

Definition 3.4 (Non-extensive process combinator). A process combinator $f \in \Sigma$ is *non-extensive* w.r.t. λ -bisimilarity metric \mathbf{d} if

$$\mathbf{d}(f(s_1, \dots, s_n), f(t_1, \dots, t_n)) \leq \max_{i=1}^n \mathbf{d}(s_i, t_i)$$

for all closed process terms $s_i, t_i \in \mathsf{T}(\Sigma)$.

Probabilistic action prefix, nondeterministic alternative composition, and probabilistic alternative composition are non-extensive w.r.t. \mathbf{d} .

Theorem 3.5. *The process combinators*

- *probabilistic action prefix* $a. \bigoplus_{i=1}^n [p_i]$
- *nondeterministic alternative composition* $_ + _$
- *probabilistic alternative composition* $_ +_p _$

are non-extensive w.r.t. λ -bisimilarity metric \mathbf{d} for any $\lambda \in (0, 1]$.

Proof. Follows directly from Proposition 3.1. □

All other operators of Σ_{PA} are not non-extensive.

Proposition 3.6. *None of the process combinators*

- *sequential composition* $_ ; _$
- *synchronous parallel composition* $_ | _$
- *asynchronous parallel composition* $_ || _$
- *CSP-like parallel composition* $_ ||_B _$
- *probabilistic parallel composition* $_ ||_p _$

is non-extensive w.r.t. λ -bisimilarity metric \mathbf{d} for any $\lambda \in (0, 1]$.

Proof. Follows directly from Propositions 3.2 and 3.3. □

We proceed now with the compositionality property of non-expansiveness.

Definition 3.7 (Non-expansive process combinator). A process combinator $f \in \Sigma$ is *non-expansive* w.r.t. λ -bisimilarity metric \mathbf{d} if

$$\mathbf{d}(f(s_1, \dots, s_n), f(t_1, \dots, t_n)) \leq \sum_{i=1}^n \mathbf{d}(s_i, t_i)$$

for all closed process terms $s_i, t_i \in \mathsf{T}(\Sigma)$.

It is clear that if a process combinator f is non-extensive, then f is non-expansive. Moreover, the two notions coincide when f is unary.

Theorem 3.8. *All non-recursive process combinators of Σ_{PA} are non-expansive w.r.t. λ -bisimilarity metric \mathbf{d} for any $\lambda \in (0, 1]$.*

Proof. Follows directly from Propositions 3.1 and 3.2 and the observation that $d^a, d_{1,2}^1 \leq d^s \leq \mathbf{d}(s_1, t_1) + \mathbf{d}(s_2, t_2)$. \square

Theorem 3.8 generalizes a similar result of [DGJP04] which considered only PTSs without non-deterministic branching and only a small set of process combinators. The analysis which operators are non-extensive (Theorem 3.5) and the tight distance bounds (Propositions 3.1, and 3.2 and 3.3) are novel.

4. RECURSIVE PROCESSES

Recursion is necessary to express infinite (non-terminating) behavior in terms of finite process expressions. Moreover, recursion allows us to express repetitive finite behavior in a compact way. We will discuss now compositional reasoning over probabilistic processes that are composed by recursive process combinators. We will see that the compositionality properties of non-extensiveness and non-expansiveness used for non-recursive process combinators (Section 3.3) fall short for recursive process combinators. We will propose the more general property of uniform continuity (Section 4.3) that captures the inherent nature of compositional reasoning over probabilistic processes. In fact, it allows us to reason compositionally over processes that are composed by both recursive and non-recursive process combinators. In the next section we apply these results to reason compositionally over a communication protocol and derive its respective performance properties. To the best of our knowledge this is the first study which explores systematically compositional reasoning over recursive processes in the context of bisimulation metric semantics. We remark that recursive process combinators are indispensable for effective modeling and verification of safety critical systems, network protocols, and systems biology.

4.1. Recursive process combinators. We define $P_{PA \cup}$ as disjoint extension of P_{PA} with the following operators:

- finite iteration $_{}^n$,
- infinite iteration $_{}^\omega$,
- binary Kleene-star iteration $_{}^*$,
- probabilistic Kleene-star iteration $_{}^{*p}$,
- finite replication $!^n _{}$,
- infinite replication (bang) operator $! _{}$, and
- probabilistic bang operator $!_p _{}$

The operational semantics of these operators is specified by the rules in Table 3.

The finite iteration t^n (resp. infinite iteration t^ω) of process t expresses that t is performed n times (resp. infinitely often) in sequel. The binary Kleene-star expresses for $t_1^*t_2$ that either t_1 is performed infinitely often in sequel, or t_1 is performed a finite number of times in sequel, followed by t_2 . The bang operator expresses for $!t$ (resp. finite replication $!^n t$) that infinitely many copies (resp. n copies) of t evolve asynchronously. The probabilistic Kleene-star iteration [Bar04, Section 5.2.4(vi)] expresses that $t_1^{*p}t_2$ evolves to a probabilistic choice (with respectively the probability p and $1 - p$) between the two nondeterministic choices of the Kleene star operation $t_1^*t_2$ for actions which can be performed by both t_1 and t_2 . For actions that can be performed by either only t_1 or only t_2 , $t_1^{*p}t_2$

$$\begin{array}{c}
\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x^{n+1} \xrightarrow{a} \mu; \delta(x^n)} \quad \frac{x \xrightarrow{\surd} \mu}{x^{n+1} \xrightarrow{\surd} \mu} \quad \frac{}{x^0 \xrightarrow{\surd} \delta(0)} \quad \frac{x \xrightarrow{\surd} \mu \quad x \xrightarrow{a} \nu \quad a \neq \surd \quad n > m}{x^n \xrightarrow{a} \nu; \delta(x^m)} \\
\\
\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x^\omega \xrightarrow{a} \mu; \delta(x^\omega)} \quad \frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x^* y \xrightarrow{a} \mu; \delta(x^* y)} \quad \frac{y \xrightarrow{a} \nu}{x^* y \xrightarrow{a} \nu} \\
\\
\frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu \quad a \neq \surd}{x^{*p} y \xrightarrow{a} \nu \oplus_p \mu; \delta(x^{*p} y)} \quad \frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu \quad a \neq \surd}{x^{*p} y \xrightarrow{a} \mu; \delta(x^{*p} y)} \quad \frac{x \xrightarrow{a} \nu \quad y \xrightarrow{a} \nu \quad a \neq \surd}{x^{*p} y \xrightarrow{a} \nu} \quad \frac{y \xrightarrow{\surd} \nu}{x^{*p} y \xrightarrow{\surd} \nu} \\
\\
\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{!^{n+1} x \xrightarrow{a} \mu \parallel \delta(!^n x)} \quad \frac{x \xrightarrow{\surd} \mu}{!^{n+1} x \xrightarrow{\surd} \mu} \quad \frac{}{!^0 x \xrightarrow{\surd} \delta(0)} \\
\\
\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{!x \xrightarrow{a} \mu \parallel \delta(!x)} \quad \frac{x \xrightarrow{a} \mu \quad a \neq \surd}{!_p x \xrightarrow{a} \mu \oplus_p (\mu \parallel \delta(!_p x))}
\end{array}$$

Table 3: Standard recursive process combinators

behaves just like $t_1^* t_2$. The probabilistic bang replication [MS13, Fig. 1] expresses that $!_p t$ replicates the argument process t with probability $1 - p$ and behave just like t with probability p .

4.2. Distance between processes combined by recursive process combinators. We develop now tight bounds on the distance between processes combined by the recursive process combinators presented in Table 3.

Proposition 4.1. *Let $P = (\Sigma, A, R)$ be any PTSS with $P_{PA \cup} \sqsubseteq P$. For all terms $s, s_i, t, t_i \in T(\Sigma)$ it holds:*

- (a) $\mathbf{d}(s^n, t^n) \leq d^n$
- (b) $\mathbf{d}(!^n s, !^n t) \leq d^n$
- (c) $\mathbf{d}(s^\omega, t^\omega) \leq d^\omega$
- (d) $\mathbf{d}(!s, !t) \leq d^!$
- (e) $\mathbf{d}(s_1^* s_2, t_1^* t_2) \leq \max(\mathbf{d}(s_1^\omega, t_1^\omega), \mathbf{d}(s_2, t_2))$
- (f) $\mathbf{d}(s_1^{*p} s_2, t_1^{*p} t_2) \leq \mathbf{d}(s_1^* s_2, t_1^* t_2)$
- (g) $\mathbf{d}(!_p s, !_p t) \leq \begin{cases} \mathbf{d}(s, t) \frac{1}{1-(1-p)(\lambda^2 - \lambda \mathbf{d}(s, t))} & \text{if } \mathbf{d}(s, t) \in (0, 1) \\ \mathbf{d}(s, t) & \text{if } \mathbf{d}(s, t) \in \{0, 1\} \end{cases}$

with

$$d^n = \begin{cases} \mathbf{d}(s, t) \frac{1-(\lambda - \mathbf{d}(s, t))^n}{1-(\lambda - \mathbf{d}(s, t))} & \text{if } \mathbf{d}(s, t) \in (0, 1) \\ \mathbf{d}(s, t) & \text{if } \mathbf{d}(s, t) \in \{0, 1\} \end{cases}$$

$$d^{!n} = \begin{cases} \mathbf{d}(s, t) \frac{1-(\lambda^2 - \lambda \mathbf{d}(s, t))^n}{1-(\lambda^2 - \lambda \mathbf{d}(s, t))} & \text{if } \mathbf{d}(s, t) \in (0, 1) \\ \mathbf{d}(s, t) & \text{if } \mathbf{d}(s, t) \in \{0, 1\} \end{cases}$$

$$d^\omega = \begin{cases} \mathbf{d}(s, t) \frac{1}{1 - (\lambda - \mathbf{d}(s, t))} & \text{if } \mathbf{d}(s, t) \in (0, 1) \\ \mathbf{d}(s, t) & \text{if } \mathbf{d}(s, t) \in \{0, 1\} \end{cases}$$

$$d^! = \begin{cases} \mathbf{d}(s, t) \frac{1}{1 - (\lambda^2 - \lambda \mathbf{d}(s, t))} & \text{if } \mathbf{d}(s, t) \in (0, 1) \\ \mathbf{d}(s, t) & \text{if } \mathbf{d}(s, t) \in \{0, 1\} \end{cases}$$

Proof. First of all we observe that $\frac{1 - (\lambda - \mathbf{d}(s, t))^n}{1 - (\lambda - \mathbf{d}(s, t))} = \sum_{k=0}^{n-1} (\lambda - \mathbf{d}(s, t))^k$ and $\frac{1 - (\lambda^2 - \lambda \mathbf{d}(s, t))^n}{1 - (\lambda^2 - \lambda \mathbf{d}(s, t))} = \sum_{k=0}^{n-1} (\lambda^2 - \lambda \mathbf{d}(s, t))^k$.

Consider first the finite iteration operator $_n$. The cases $\mathbf{d}(s, t) = 0$ and $\mathbf{d}(s, t) = 1$ are immediate. Consider the case $0 < \mathbf{d}(s, t) < 1$. The proof obligation can be rewritten as $\mathbf{d}(s^n, t^n) \leq \mathbf{d}(s, t) \sum_{k=0}^{n-1} (\lambda - \mathbf{d}(s, t))^k$. We reason by induction over n . The base case $n = 0$ is immediate. Let us consider the inductive step $n + 1$. By the rules in Tables 1–3, we infer that s^{n+1} is bisimilar to $s; s^n$ (i.e. they are in bisimulation distance 0) and that t^{n+1} is bisimilar to $t; t^n$. Hence $\mathbf{d}(s^{n+1}, t^{n+1}) = \mathbf{d}(s; s^n, t; t^n)$. By Proposition 3.2.(a) we have $\mathbf{d}(s; s^n, t; t^n) \leq \mathbf{d}(s, t) + \mathbf{d}(s^n, t^n)(\lambda - \mathbf{d}(s, t)) =$ (by the inductive hypothesis over n) $\mathbf{d}(s, t) + (\mathbf{d}(s, t) \sum_{k=0}^{n-1} (\lambda - \mathbf{d}(s, t))^k)(\lambda - \mathbf{d}(s, t)) = \mathbf{d}(s, t) \sum_{k=0}^n (\lambda - \mathbf{d}(s, t))^k$. Summarizing, $\mathbf{d}(s^{n+1}, t^{n+1}) \leq \mathbf{d}(s, t) \sum_{k=0}^n (\lambda - \mathbf{d}(s, t))^k$, thus confirming the thesis.

Consider now the finite replication operator $!^n$. The cases $\mathbf{d}(s, t) = 1$ and $\mathbf{d}(s, t) = 0$ are immediate. Consider the case $0 < \mathbf{d}(s, t) < 1$. The proof obligation can be rewritten as $\mathbf{d}(s, !^n t) \leq \mathbf{d}(s, t) \sum_{k=0}^{n-1} (\lambda^2 - \lambda \mathbf{d}(s, t))^k$. We reason by induction over n . The base case $n = 0$ is immediate. Let us consider the inductive step $n + 1$. By the rules in Tables 1–3, we infer that $!^{n+1} s$ is bisimilar to $s \ ||| !^n s$ and that $!^{n+1} t$ is bisimilar to $t \ ||| !^n t$. Hence $\mathbf{d}(s, !^{n+1} t) = \mathbf{d}(s \ ||| !^n s, t \ ||| !^n t)$. By Proposition 3.2.(c) we get $\mathbf{d}(s \ ||| !^n s, t \ ||| !^n t) \leq \mathbf{d}(s, t) + (\lambda^2 - \lambda \mathbf{d}(s, t)) \mathbf{d}(s, !^n t) \leq$ (inductive hypothesis over n) $\mathbf{d}(s, t) + (\lambda^2 - \lambda \mathbf{d}(s, t)) \mathbf{d}(s, t) (\sum_{k=0}^{n-1} (\lambda^2 - \lambda \mathbf{d}(s, t))^k) = \mathbf{d}(s, t) \sum_{k=0}^n (\lambda^2 - \lambda \mathbf{d}(s, t))^k$. Summarizing, we have $\mathbf{d}(s, !^{n+1} t) \leq \mathbf{d}(s, t) \sum_{k=0}^n (\lambda^2 - \lambda \mathbf{d}(s, t))^k$. This confirms the thesis.

Consider the infinite iteration operator $_^\omega$. The cases $\mathbf{d}(s, t) = 1$ and $\mathbf{d}(s, t) = 0$ are immediate. Consider the case $0 < \mathbf{d}(s, t) < 1$. By the rules in Tables 1–3, we infer that s^ω is bisimilar to $s; s^\omega$ and that t^ω is bisimilar to $t; t^\omega$. Hence $\mathbf{d}(s^\omega, t^\omega) = \mathbf{d}(s; s^\omega, t; t^\omega)$. By Proposition 3.2.(a) we get $\mathbf{d}(s; s^\omega, t; t^\omega) \leq \mathbf{d}(s, t) + (\lambda - \mathbf{d}(s, t)) \mathbf{d}(s^\omega, t^\omega)$. Hence we have $\mathbf{d}(s^\omega, t^\omega) \leq \mathbf{d}(s, t) + (\lambda - \mathbf{d}(s, t)) \mathbf{d}(s^\omega, t^\omega)$, from which we infer $\mathbf{d}(s^\omega, t^\omega) \leq \mathbf{d}(s, t) \frac{1}{1 - (\lambda - \mathbf{d}(s, t))} = d^\omega$.

Consider now the bang operator $!_$. The cases $\mathbf{d}(s, t) = 1$ and $\mathbf{d}(s, t) = 0$ are immediate. Consider the case $0 < \mathbf{d}(s, t) < 1$. By the rules in Tables 1–3, we infer that $!s$ is bisimilar to $s \ ||| !s$ and that $!t$ is bisimilar to $t \ ||| !t$. Hence $\mathbf{d}(s, !t) = \mathbf{d}(s \ ||| !s, t \ ||| !t)$. By Proposition 3.2.(c) we get $\mathbf{d}(s \ ||| !s, t \ ||| !t) \leq \mathbf{d}(s, t) + (\lambda^2 - \lambda \mathbf{d}(s, t)) \mathbf{d}(s, !t)$. Hence we have $\mathbf{d}(s, !t) \leq \mathbf{d}(s, t) + (\lambda^2 - \lambda \mathbf{d}(s, t)) \mathbf{d}(s, !t)$, from which we infer $\mathbf{d}(s, !t) \leq \mathbf{d}(s, t) \frac{1}{1 - (\lambda^2 - \lambda \mathbf{d}(s, t))} = d^!$.

Consider the binary Kleene star operator $_*$. Observe that the term $s_1^* s_2$ is bisimilar to $(s_1; (s_1^* s_2)) + s_2$ and that the term $t_1^* t_2$ is bisimilar to $(t_1; (t_1^* t_2)) + t_2$. Proposition 3.1.(b) shows $\mathbf{d}(s_1^* s_2, t_1^* t_2) = \mathbf{d}((s_1; (s_1^* s_2)) + s_2, (t_1; (t_1^* t_2)) + t_2) = \max\{\mathbf{d}((s_1; (s_1^* s_2)), (t_1; (t_1^* t_2))), \mathbf{d}(s_2, t_2)\}$. If $\max\{\mathbf{d}((s_1; (s_1^* s_2)), (t_1; (t_1^* t_2))), \mathbf{d}(s_2, t_2)\} = \mathbf{d}((s_1; (s_1^* s_2)), (t_1; (t_1^* t_2)))$, we get $\mathbf{d}(s_1^* s_2, t_1^* t_2) = \mathbf{d}((s_1; (s_1^* s_2)), (t_1; (t_1^* t_2)))$, where, by Proposition 3.2.(a), $\mathbf{d}((s_1; (s_1^* s_2)), (t_1; (t_1^* t_2))) = \mathbf{d}(s_1, t_1) + (\lambda - \mathbf{d}(s_1, t_1)) \mathbf{d}(s_1^* s_2, t_1^* t_2)$, thus giving $\mathbf{d}(s_1^* s_2, t_1^* t_2) = \mathbf{d}(s_1, t_1) \frac{1}{1 - (\lambda - \mathbf{d}(s_1, t_1))}$. Therefore we conclude that $\mathbf{d}(s_1^* s_2, t_1^* t_2) = \max\{\mathbf{d}(s_1, t_1) \frac{1}{1 - (\lambda - \mathbf{d}(s_1, t_1))}, \mathbf{d}(s_2, t_2)\} = \max\{\mathbf{d}(s_1^\omega, t_1^\omega), \mathbf{d}(s_2, t_2)\}$. This confirms the thesis.

Consider now the probabilistic Kleene star operator. The second, third and fourth rule specifying the probabilistic Kleene star operator define the same operational behavior as the nondeterministic Kleene star operator. Since the target of the first rule for the probabilistic Kleene star operator is a convex combination of the targets of the second and the third rule, the thesis follows.

Consider now the probabilistic bang operator. The bound on the distance of processes composed by the probabilistic bang operator can be understood by observing that the term $!_p s$ behaves as $!^{n+1} s$ with probability $p(1-p)^n$. Hence, by Proposition 4.1.(b) we get $\mathbf{d}(!_p s, !_p t) \leq \sum_{n=0}^{\infty} p(1-p)^n \mathbf{d}(!^{n+1} s, !^{n+1} t) \leq \sum_{n=0}^{\infty} p(1-p)^n d^{n+1} = \mathbf{d}(s, t) / (1 - (1-p)(\lambda^2 - \lambda \mathbf{d}(s, t)))$. \square

The bounds for the combinators in Proposition 4.1 are immediate when the distance between the process arguments is either 0 or 1. We explain those bounds by assuming that the distance between the process arguments is neither 0 nor 1.

First we explain the distance bounds for the nondeterministic recursive process combinators. To understand the distance bound between processes that iterate finitely often (Proposition 4.1.(a)), observe that s^n and $s; \dots; s$, with $s; \dots; s$ denoting n sequentially composed instances of s , denote the same PTSs (up to renaming of states). Recursive application of the distance bound for operator $_-; _-$ (Proposition 3.2.(a)) yields $\mathbf{d}(s^n, t^n) = \mathbf{d}(s; \dots; s, t; \dots; t) \leq \mathbf{d}(s, t) \sum_{k=0}^{n-1} (\lambda - \mathbf{d}(s, t))^k = d^n$. The same reasoning applies to the finite replication operator (Proposition 4.1.(b)) by observing that $!^n s$ and $s \parallel \dots \parallel s$, with $s \parallel \dots \parallel s$ denoting n occurrences of s that evolve asynchronously, denote the same PTSs (up to renaming of states), thus giving $\mathbf{d}(!^n s, !^n t) = \mathbf{d}(s \parallel \dots \parallel s, t \parallel \dots \parallel t) \leq \mathbf{d}(s, t) \sum_{k=0}^{n-1} (\lambda^2 - \lambda \mathbf{d}(s, t))^k = d^n$.

The distance between processes that may iterate infinitely many times (Proposition 4.1.(c)), and the distance between processes that may spawn infinitely many copies that evolve asynchronously (Proposition 4.1.(d)) are the limit of the respective finite iteration and replication bounds. The distance between the Kleene-star iterated processes $s_1^* s_2$ and $t_1^* t_2$ (Proposition 4.1.(e)) is bounded by the maximum of the distance $\mathbf{d}(s_1^\omega, t_1^\omega)$ (infinite iteration of s_1 and t_1 s.t. s_2 and t_2 never evolve), and the distance $\mathbf{d}(s_2, t_2)$ (s_2 and t_2 evolve immediately). The case where s_1 and t_1 iterate n -times and then s_2 and t_2 evolve leads always to a distance $\mathbf{d}(s_1^n, t_1^n) + (\lambda - \mathbf{d}(s_1, t_1))^n \mathbf{d}(s_2, t_2) \leq \max(\mathbf{d}(s_1^\omega, t_1^\omega), \mathbf{d}(s_2, t_2))$.

Now we explain the bounds for the probabilistic recursive process combinators. The distance between processes composed by the probabilistic Kleene star is bounded by the distance between those processes composed by the nondeterministic Kleene star (Proposition 4.1.(f)), since the second, the third and the fourth rule specifying the probabilistic Kleene star define the same operational behavior as the nondeterministic Kleene star, and the first rule which defines a convex combination of these transitions applies only for those actions that both of the combined processes can perform. In fact, $\mathbf{d}(s_1^{*p} s_2, t_1^{*p} t_2) = \mathbf{d}(s_1^* s_2, t_1^* t_2)$ if the initial actions that can be performed by processes s_1, t_1 are disjoint from the initial actions that can be performed by processes s_2, t_2 (and hence the first rule defining $_-^{*p} _-$ cannot be applied). Thus, the distance bound of the probabilistic Kleene star coincides with the distance bound of the nondeterministic Kleene star. The bound on the distance of processes composed by the probabilistic bang operator can be understood by observing that $!_p s$ behaves as $!^{n+1} s$ with probability $p(1-p)^n$. Hence, by Proposition 4.1.(b) we get $\mathbf{d}(!_p s, !_p t) \leq \sum_{n=0}^{\infty} p(1-p)^n \mathbf{d}(!^{n+1} s, !^{n+1} t) \leq \sum_{n=0}^{\infty} p(1-p)^n d^{n+1} = \mathbf{d}(s, t) / (1 - (1-p)(\lambda^2 - \lambda \mathbf{d}(s, t)))$.

The distance bounds on the distance between processes composed by recursive process combinators (Proposition 4.1) are tight.

Proposition 4.2. *Let $\epsilon_i \in [0, 1]$. There are processes $s_i, t_i \in T(\Sigma_{PA})$ with $\mathbf{d}(s_i, t_i) = \epsilon_i$ such that the inequalities in Proposition 4.1 become equalities.*

Proof. The witness processes of Proposition 3.3 that were used to show that the inequality in Proposition 3.2.(a) becomes an equality, suffice for Propositions 4.1.(a), 4.1.(c), 4.1.(e), 4.1.(f). The witness processes of Proposition 3.3 that were used to show that the inequality in Proposition 3.2.(c) becomes an equality, suffice for Propositions 4.1.(b), 4.1.(d), 4.1.(g). \square

4.3. Compositional reasoning over recursive processes. From Propositions 4.1 and 4.2 it follows that none of the recursive process combinators discussed in this section satisfies the compositionality property of non-expansiveness.

Proposition 4.3. *None of the recursive process combinators of Σ_{PA^\cup} (unbounded recursion and bounded recursion with $n \geq 2$) is non-expansive w.r.t. λ -bisimilarity metric \mathbf{d} for any $\lambda \in (0, 1]$.*

Proof. Follows directly from Propositions 4.1 and 4.2 and the observation that $d^\omega \geq d^!$, $d^n \geq d^m > \mathbf{d}(s, t)$ whenever $0 < \mathbf{d}(s, t) < 1$. \square

However, a weaker property suffices to facilitate compositional reasoning. To reason compositionally over probabilistic processes it is enough if the distance between the composed processes can be related to the distance between their parts. In essence, compositional reasoning over probabilistic processes is possible whenever a small variance in the behavior of the parts leads to a bounded small variance in the behavior of the composed processes.

We introduce uniform continuity as the compositionality property for both recursive and non-recursive process combinators. Uniform continuity generalizes the properties non-extensiveness and non-expansiveness for non-recursive process combinators.

Definition 4.4 (Uniformly continuous process combinator). A process combinator $f \in \Sigma$ is *uniformly continuous* w.r.t. λ -bisimilarity metric \mathbf{d} if for all $\epsilon > 0$ there are $\delta_1, \dots, \delta_n > 0$ such that

$$\forall i = 1, \dots, n. \mathbf{d}(s_i, t_i) < \delta_i \implies \mathbf{d}(f(s_1, \dots, s_n), f(t_1, \dots, t_n)) < \epsilon$$

for all closed process terms $s_i, t_i \in \mathbb{T}(\Sigma)$.

Note that by definition each non-expansive operator is also uniformly continuous (by $\delta_i = \epsilon/n$). A uniformly continuous combinator f ensures that for any non-zero bisimulation distance ϵ there are appropriate non-zero bisimulation distances δ_i s.t. for any composed process $f(s_1, \dots, s_n)$ the distance to the composed process where each s_i is replaced by any t_i with $\mathbf{d}(s_i, t_i) < \delta_i$ is $\mathbf{d}(f(s_1, \dots, s_n), f(t_1, \dots, t_n)) < \epsilon$. We consider the uniform notion of continuity (technically, the δ_i depend only on ϵ and are independent of the concrete states s_i) because we aim at universal compositionality guarantees.

A particular case of uniform continuity is Lipschitz continuity, which requires that there is a constant $K \in \mathbb{R}_{\geq 0}$ such that $\delta_i = \epsilon/(n \cdot K)$. Intuitively, this ensures that the distance between the composed processes is limited in how fast it can change due to the change of the distance between the components.

Definition 4.5 (Lipschitz continuous process combinator). A process combinator $f \in \Sigma$ is *Lipschitz continuous* w.r.t. λ -bisimilarity metric \mathbf{d} if there exists a constant $K \in \mathbb{R}_{\geq 0}$ with

$$\mathbf{d}(f(s_1, \dots, s_n), f(t_1, \dots, t_n)) \leq K \sum_{i=1}^n \mathbf{d}(s_i, t_i)$$

for all closed process terms $s_i, t_i \in \mathbb{T}(\Sigma)$.

We refer to the constant K in Definition 4.5 as the *Lipschitz factor* for combinator f , and we may say that f is *K -Lipschitz continuous*. Note that by definition a non-expansive operator is Lipschitz continuous (by $K = 1$) and a Lipschitz continuous operator is uniformly continuous (by $\delta_i = \epsilon/(n \cdot K)$).

The distance bounds of Section 4.2 allow us to derive that finitely recursing process combinators are Lipschitz continuous (and therefore also uniformly continuous) w.r.t. both non-discounted and discounted bisimilarity metric (Theorem 4.6). On the contrary, unbounded recursing process combinators are Lipschitz continuous and uniformly continuous only w.r.t. discounted bisimilarity metric (Theorem 4.7 and Proposition 4.8).

Theorem 4.6. *The process combinators*

- finite iteration $_n$
- finite replication $\!^n_$
- probabilistic replication (bang) $\!_p_$

are Lipschitz continuous w.r.t. λ -bisimilarity metric \mathbf{d} for any $\lambda \in (0, 1]$.

Proof. For finite iteration operator, this follows directly from Propositions 4.1.(a) and the observation that $\frac{1-(\lambda-\mathbf{d}(s,t))^n}{1-(\lambda-\mathbf{d}(s,t))} \leq n = K$. For finite replication operator, this follows directly from Propositions 4.1.(b) and the observation that $\frac{1-(\lambda^2-\lambda\mathbf{d}(s,t))^n}{1-(\lambda^2-\lambda\mathbf{d}(s,t))} \leq n = K$. For the probabilistic bang operator it follows from Proposition 4.1.(g) and the observation that $\frac{1}{1-(1-p)(\lambda^2-\lambda\mathbf{d}(s,t))} \leq \frac{1}{1-(1-p)\lambda^2} = K$. \square

Note that the probabilistic bang operator is Lipschitz continuous w.r.t. non-discounted bisimilarity metric \mathbf{d} with $\lambda = 1$ because in each step there is a non-zero probability that the process is not copied. On the contrary, the process $s_1 \!^*p s_2$ applying the probabilistic Kleene star creates with probability 1 a copy of s_1 for actions that s_1 can and s_2 cannot perform. Hence, the probabilistic Kleene star operator $_ \!^*p _$ is uniformly continuous only for discounted bisimilarity metric with $\lambda < 1$.

Theorem 4.7. *The process combinators*

- infinite iteration $_\omega$
- nondeterministic Kleene-star iteration $__*$
- probabilistic Kleene-star iteration $__ \!^*p _$, and
- infinite replication (bang) $\!_\omega$

are Lipschitz continuous w.r.t. discounted λ -bisimilarity metric \mathbf{d} for any $\lambda \in (0, 1)$.

Proof. For infinite iteration, nondeterministic Kleene star iteration and probabilistic Kleene star iteration this follows by Proposition 4.1.(c), 4.1.(e), 4.1.(f) and the observation that $\frac{1}{1-\lambda} \leq \frac{1}{1-\lambda-\mathbf{d}(s,t)} \leq \frac{1}{1-\lambda} = K$. For infinite replication this follows by Proposition 4.1.(d) and the observation that $\frac{1}{1-(\lambda^2-\lambda\mathbf{d}(s,t))} \leq \frac{1}{1-\lambda^2} = K$. \square

Proposition 4.8. *None of the process combinators*

- infinite iteration $_\omega$
- nondeterministic Kleene-star iteration $__*$
- probabilistic Kleene-star iteration $__ \!^*p _$, and
- infinite replication (bang) $\!_\omega$

is uniformly continuous w.r.t. the non-discounted λ -bisimilarity metric \mathbf{d} with $\lambda = 1$.

Proof. Follows directly from Propositions 4.1 and 4.2. We will reason in detail for the first case of infinite iteration operator. Let ϵ be any fixed real with $0 < \epsilon < 1$. We will show that there

is no $\delta > 0$ s.t. for all $s, t \in \mathsf{T}(\Sigma)$ with $\mathbf{d}(s, t) < \delta$ we have $\mathbf{d}(s^\omega, t^\omega) < \epsilon$. We will show this by contradiction. Assume there is some $\delta > 0$. Consider $s = a.([1 - \delta/2]\epsilon \oplus [\delta/2]0)$ and $t = a.\epsilon$. We have $\mathbf{d}(s, t) = \delta/2 < \delta$ and $\mathbf{d}(s^\omega, t^\omega) = 1 > \epsilon$. Contradiction. Similar reasoning applies also to the other process combinators. \square

Note that the processes used in the proof of Proposition 4.8 are witnesses that these combinators are not continuous at all.

Given any discount factor λ , all process combinators discussed so far that are uniformly continuous wrt. λ -bisimilarity metric \mathbf{d} are also Lipschitz continuous wrt. \mathbf{d} . We conclude this section by discussing the copy operator cp of [BIM95, FvGdW12] as an example of an operator being uniformly continuous but not Lipschitz continuous wrt. discounted λ -bisimilarity metric \mathbf{d} with any $\lambda \in (0, 1)$.

The copy operator cp is defined by the rules

$$\frac{x \xrightarrow{a} \mu}{\mathsf{cp}(x) \xrightarrow{a} \mu} (a \notin \{l, r\}) \qquad \frac{x \xrightarrow{l} \mu \quad x \xrightarrow{r} \nu}{\mathsf{cp}(x) \xrightarrow{s} \mathsf{cp}(\mu) \mid \mathsf{cp}(\nu)}$$

The copy operator cp specifies the fork operation of operating systems. Actions l and r are the left and right *forking actions*, and s is the resulting *split action*. The fork of t is the process $\mathsf{cp}(t)$ evolving by t to the parallel composition of the left fork (l -derivative of t) and the right fork (r -derivative of t). For all other actions $a \notin \{l, r\}$ the process $\mathsf{cp}(t)$ mimics the behavior of t .

Proposition 4.9. *The copy operator cp is not Lipschitz continuous wrt. λ -bisimilarity metric \mathbf{d} for any $\lambda \in (0, 1]$.*

Proof. Assume any discount factor $\lambda \in (0, 1]$. For any constant $L \in \mathbb{R}_{\geq 0}$, we provide suitable CCS processes s and t s.t. $\mathbf{d}(\mathsf{cp}(s), \mathsf{cp}(t)) > L\mathbf{d}(s, t)$. Let $s_1 = l.([1 - \epsilon]a \oplus [\epsilon]0) + r.([1 - \epsilon]a \oplus [\epsilon]0)$ and $t_1 = l.a + r.a$, and $s_{k+1} = l.s_k + r.s_k$ and $t_{k+1} = l.t_k + r.t_k$. Clearly $\mathbf{d}(s_k, t_k) = \lambda^k \epsilon$. Then $\mathbf{d}(\mathsf{cp}(s_k), \mathsf{cp}(t_k)) = \lambda^k(1 - (1 - \epsilon)^{2^k})$. Hence, for any k with $2^k > L$, $\mathbf{d}(\mathsf{cp}(s), \mathsf{cp}(t))/\mathbf{d}(s, t) = (1 - (1 - \epsilon)^{2^k})/\epsilon > L$ holds for $s = s_k, t = t_k$ and all $0 < \epsilon < (2^k - L)/(2^{k-1}(2^k - 1))$. Thus, the copy operator is not Lipschitz continuous wrt. λ -bisimilarity metric \mathbf{d} . \square

To prove that the copy operator cp is uniformly continuous wrt. discounted λ -bisimilarity metric \mathbf{d} with any $\lambda \in (0, 1)$, we need some preliminary results. First we show that the behavioral distance between two arbitrary terms s and t can be divided in the distance observable by the first k steps and the distance observable after step k . The step discount λ allows us to give the upper bound λ^k on the distance observable after step k .

Proposition 4.10. *Let $P = (\Sigma, A, R)$ be a PTSS and $s, t \in \mathsf{T}(\Sigma)$ arbitrary closed terms. Then*

$$\mathbf{d}(s, t) \leq \mathbf{d}_k(s, t) + \lambda^k$$

for all $k \in \mathbb{N}$.

Proof. By induction. Case $k = 0$ is trivial since $\lambda^0 = 1$. Let $(\mathbf{d} - \epsilon): \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \rightarrow [0, \epsilon]$ with $\epsilon \in [0, 1]$ be the function defined by $(\mathbf{d} - \epsilon)(s, t) = \max(\mathbf{d}(s, t) - \epsilon, 0)$. For the induction step, assume $\mathbf{d}_k \sqsupseteq \mathbf{d} - \lambda^k$. It remains to show $\mathbf{d}_{k+1} \sqsupseteq \mathbf{d} - \lambda^{k+1}$. We reason as follows:

$$\begin{aligned} & \mathbf{d}_{k+1}(s, t) \\ &= \sup_{a \in A} \{ \mathbf{H}(\lambda \cdot \mathbf{K}(\mathbf{d}_k))(der(s, a), der(t, a)) \} \\ &\geq \sup_{a \in A} \{ \mathbf{H}(\lambda \cdot \mathbf{K}(\mathbf{d} - \lambda^k))(der(s, a), der(t, a)) \} \end{aligned}$$

$$\begin{aligned} &\geq \sup_{a \in A} \{\mathbf{H}(\lambda \cdot \mathbf{K}(\mathbf{d}))(\text{der}(s, a), \text{der}(t, a))\} - \lambda^{k+1} \\ &= \mathbf{d}(s, t) - \lambda^{k+1} \end{aligned}$$

by using the properties

$$\begin{aligned} \mathbf{K}(d) &\sqsupseteq \mathbf{K}(d') && \text{if } d \sqsupseteq d' \\ \mathbf{H}(d) &\sqsupseteq \mathbf{H}(d') && \text{if } d \sqsupseteq d' \\ \mathbf{K}(d - \epsilon)(\pi, \pi') &\geq \mathbf{K}(d)(\pi, \pi') - \epsilon \\ \mathbf{H}(d - \epsilon)(\pi, \pi') &\geq \mathbf{H}(d)(\pi, \pi') - \epsilon \end{aligned} \tag{4.1}$$

for any pseudometrics d, d' and any $\epsilon \in [0, 1]$, definition of \mathbf{d}_{k+1} applied in step 1, induction hypothesis applied in step 2, the fixpoint property of bisimulation metric $\mathbf{d}(s, t) = \sup_{a \in A} \{\mathbf{H}(\lambda \cdot \mathbf{K}(\mathbf{d}))(\text{der}(s, a), \text{der}(t, a))\}$ applied in step 4, and properties of Equation 4.1 applied in steps 2 and 3. \square

Now we show that an operator is uniformly continuous w.r.t. the discounted λ -bisimilarity metric \mathbf{d} if this operator is Lipschitz continuous wrt. all up-to- k λ -bisimilarity metrics \mathbf{d}_k .

Theorem 4.11. *Let $P = (\Sigma, A, R)$ be a PTSS and $\lambda < 1$. If an operator $f \in \Sigma$ is Lipschitz continuous wrt. \mathbf{d}_k for each $k \in \mathbb{N}$, then f is uniformly continuous wrt. \mathbf{d} .*

Proof. Assume that $f \in \Sigma$ is any n -ary operator. We prove that for any $\epsilon > 0$ there exist $\delta_1, \dots, \delta_n > 0$ such that $\mathbf{d}(f(s_1, \dots, s_n), f(t_1, \dots, t_n)) < \epsilon$ whenever $\mathbf{d}(s_i, t_i) < \delta_i$ for all $i = 1, \dots, n$. Let $L_k \in \mathbb{R}_{\geq 0}$ be the Lipschitz factor for f wrt. \mathbf{d}_k , i.e.

$$\mathbf{d}_k(f(s_1, \dots, s_n), f(t_1, \dots, t_n)) \leq L_k \sum_{i=1}^n \mathbf{d}_k(s_i, t_i).$$

Together with Proposition 4.10 and property $\mathbf{d}_k \sqsubseteq \mathbf{d}$ we get

$$\mathbf{d}(f(s_1, \dots, s_n), f(t_1, \dots, t_n)) \leq L_k \sum_{i=1}^n \mathbf{d}(s_i, t_i) + \lambda^k \tag{4.2}$$

for all $k \in \mathbb{N}$. Since $\lambda < 1$, there is some $m \in \mathbb{N}$ s.t. $\lambda^m < \epsilon$. Let $\delta_i \in (0, 1]$ be such that

$$\delta_i < \frac{\epsilon - \lambda^m}{n \cdot L_m}$$

If we take $\mathbf{d}(s_i, t_i) < \delta_i$ for all $i = 1, \dots, n$ then we get

$$\begin{aligned} &\mathbf{d}(f(s_1, \dots, s_n), f(t_1, \dots, t_n)) \\ &\leq L_m \sum_{i=1}^n \mathbf{d}(s_i, t_i) + \lambda^m && \text{(Equation 4.2)} \\ &< L_m \sum_{i=1}^n \delta_i + \lambda^m \\ &\leq L_m \sum_{i=1}^n \frac{\epsilon - \lambda^m}{n \cdot L_m} + \lambda^m \\ &= \epsilon \end{aligned}$$

thus concluding that that f is uniformly continuous w.r.t. \mathbf{d} . \square

Now we show that the copy operator cp is Lipschitz-continuous wrt. the (not necessarily discounted) up-to- k λ -bisimilarity metric \mathbf{d}_k for any $k \geq 0$ and $\lambda \in (0, 1]$. Together with Theorem 4.11 this allows us to derive that cp is uniformly continuous wrt. the discounted λ -bisimilarity metric \mathbf{d} for any $\lambda \in (0, 1)$.

Proposition 4.12. *The copy operator cp is Lipschitz continuous wrt. the up-to- k λ -bisimilarity metric \mathbf{d}_k for any $k \geq 0$ and $\lambda \in (0, 1]$.*

Proof. For all $k \geq 0$, we show that the operator cp is 2^k -Lipschitz continuous wrt. the up-to- k λ -bisimilarity metric \mathbf{d}_k , namely

$$\mathbf{d}_k(\text{cp}(s), \text{cp}(t)) \leq 2^k \mathbf{d}_k(s, t)$$

holds for arbitrary terms $s, t \in \mathsf{T}(\Sigma)$. We proceed by induction over k . The base case $k = 0$ is immediate. Consider the inductive step $k + 1$. The subcase $\mathbf{d}_{k+1}(s, t) = 1$ is immediate. Consider the subcase $\mathbf{d}_{k+1}(s, t) < 1$. We consider now the two different rules specifying the copy operator and show that in each case whenever $\text{cp}(s) \xrightarrow{a} \pi$ is derivable by some of the rules then there is a transition $\text{cp}(t) \xrightarrow{a} \pi'$ derivable by the same rule s.t. $\lambda \cdot \mathbf{K}(\mathbf{d}_k)(\pi, \pi') \leq 2^{k+1} \mathbf{d}_{k+1}(s, t)$, thus confirming the thesis.

- (1) Assume that $\text{cp}(s) \xrightarrow{a} \pi$ is derived by $s \xrightarrow{a} \pi$ with $a \in A \setminus \{l, r\}$. Since $\mathbf{d}_{k+1}(s, t) < 1$ and \mathbf{d}_{k+1} satisfies the transfer condition of the bisimulation metrics, there exists a transition $t \xrightarrow{a} \pi'$ for a distributions π' with $\lambda \cdot \mathbf{K}(\mathbf{d}_k)(\pi, \pi') \leq \mathbf{d}_{k+1}(s, t)$. Finally, from $t \xrightarrow{a} \pi'$ we derive $\text{cp}(t) \xrightarrow{a} \pi'$.
- (2) Assume that $\text{cp}(s) \xrightarrow{a} \pi$ is derived by $s \xrightarrow{l} \pi_1$ and $s \xrightarrow{r} \pi_2$ with $a = s$ and $\pi = \text{cp}(\pi_1) \mid \text{cp}(\pi_2)$. Since $\mathbf{d}_{k+1}(s, t) < 1$ and \mathbf{d}_{k+1} satisfies the transfer condition of the bisimulation metrics, there exist transitions $t \xrightarrow{l} \pi'_1$ and $t \xrightarrow{r} \pi'_2$ for distributions π'_1, π'_2 with $\lambda \cdot \mathbf{K}(\mathbf{d}_k)(\pi_1, \pi'_1) \leq \mathbf{d}_{k+1}(s, t)$ and $\lambda \cdot \mathbf{K}(\mathbf{d}_k)(\pi_2, \pi'_2) \leq \mathbf{d}_{k+1}(s, t)$. From $t \xrightarrow{l} \pi'_1$ and $t \xrightarrow{r} \pi'_2$ we derive $\text{cp}(t) \xrightarrow{s} \text{cp}(\pi'_1) \mid \text{cp}(\pi'_2)$. Finally we have

$$\begin{aligned} & \lambda \mathbf{K}(\mathbf{d}_k)(\text{cp}(\pi_1) \mid \text{cp}(\pi_2), \text{cp}(\pi'_1) \mid \text{cp}(\pi'_2)) \\ & \leq \lambda(1 - (1 - \mathbf{K}(\mathbf{d}_k)(\text{cp}(\pi_1), \text{cp}(\pi'_1)))(1 - \mathbf{K}(\mathbf{d}_k)(\text{cp}(\pi_2), \text{cp}(\pi'_2)))) \\ & \leq \lambda(\mathbf{K}(\mathbf{d}_k)(\text{cp}(\pi_1), \text{cp}(\pi'_1)) + \mathbf{K}(\mathbf{d}_k)(\text{cp}(\pi_2), \text{cp}(\pi'_2))) \\ & \leq \lambda(2^k \mathbf{K}(\mathbf{d}_k)(\pi_1, \pi'_1) + 2^k \mathbf{K}(\mathbf{d}_k)(\pi_2, \pi'_2)) \\ & \leq \lambda(2^k \mathbf{d}_{k+1}(s, t)/\lambda + 2^k \mathbf{d}_{k+1}(s, t)/\lambda) \\ & = 2^{k+1} \mathbf{d}_{k+1}(s, t) \end{aligned}$$

with the first step by the inductive hypothesis and Theorem 2.20 (using the fact that the candidate modulus of continuity of operator \mid given by $z(\epsilon_1, \epsilon_2) = \lambda[1 - (1 - \epsilon_1/\lambda)(1 - \epsilon_2/\lambda)]$ is concave), the third step again by the inductive hypothesis and by Theorem 2.20 (using the fact that the candidate modulus of continuity of operator cp given by $z(\epsilon) = 2^k \epsilon$ is concave). □

Theorem 4.13. *The copy operator cp is uniformly continuous wrt. the discounted λ -bisimilarity metric \mathbf{d} for any $\lambda \in (0, 1)$.*

Proof. Directly by Proposition 4.12 and Theorem 4.11. □

$BRP(N, T, p, q) = RC(N, T, p, q) \parallel_B TV$, where $B = \{c(d, b) \mid d \in D, b \in \{0, 1\}\} \cup \{ack, lost\}$

$$\begin{aligned}
 RC(N, T, p, q) &= \left[\sum_{1 \leq n \leq N, n=2k} i(n) \cdot \left(CH(0, T, p, q); CH(1, T, p, q) \right)^{\frac{n}{2}} \right. \\
 &\quad + \\
 &\quad \left. \sum_{1 \leq n \leq N, n=2k+1} i(n) \cdot \left(\left(CH(0, T, p, q); CH(1, T, p, q) \right)^{\frac{n-1}{2}}; CH(0, T, p, q) \right) \right]; \\
 &\quad res(OK). \varepsilon \\
 CH(b, t, p, q) &= \sum_{d \in D} i(d) \cdot CH'(d, b, t, p, q) \\
 CH'(d, b, t, p, q) &= \begin{cases} (\perp. CH'(d, b, t-1, p, q)) \oplus_p (c(d, b). CH_2(d, b, t, p, q)) & \text{if } t > 0 \\ res(NOK) & \text{if } t = 0 \end{cases} \\
 CH_2(d, b, t, p, q) &= \begin{cases} (lost. CH'(d, b, t-1, p, q)) \oplus_q (ack. \varepsilon) & \text{if } t > 0 \\ res(NOK) & \text{if } t = 0 \end{cases} \\
 TV &= \left[\left(\left(\sum_{d \in D} c(d, 1). (ack. \varepsilon + lost. \varepsilon) \right)^* \left(\sum_{d \in D} c(d, 0). o(d). (ack. \varepsilon + lost. \varepsilon) \right) \right); \right. \\
 &\quad \left. \left(\left(\sum_{d \in D} c(d, 0). (ack. \varepsilon + lost. \varepsilon) \right)^* \left(\sum_{d \in D} c(d, 1). o(d). (ack. \varepsilon + lost. \varepsilon) \right) \right) \right]^\omega
 \end{aligned}$$

Figure 1: Specification of the Bounded Retransmission Protocol

5. APPLICATION

To advocate both uniform continuity as adequate property for compositional reasoning as well as bisimulation metric semantics as a suitable distance measure for performance validation of communication protocols, we exemplify the discussed compositional reasoning method by analyzing the bounded retransmission protocol (BRP) as a case study.

The BRP allows us to transfer streams of data from a sender (e.g. a remote control RC) to a receiver (e.g. a TV). The RC tries to send to the TV a stream of n data, d_0, \dots, d_{n-1} , with each d_i a member of the finite data domain D . The length n of the stream is bounded by a given N . Each datum d_i is sent separately and has probability p to get lost. When the TV receives a datum d_i , it sends back an acknowledgment message to the RC, which may also get lost, with probability q . If the RC does not receive the acknowledgment for datum d_i within a given time bound, it assumes that d_i got lost and retries to transmit it. However, the maximal number of attempts for d_i is a given T , meaning that T failures for any datum d_i imply the failure of the whole transmission. Since also the acknowledgment message may get lost, it may happen that the RC sends more than once the same datum d_i notwithstanding that it was correctly received by the TV. Therefore, the RC attaches a control bit b to each datum d_i that it sends to the TV, s.t. the TV can recognize if this datum is

original or already received. Data items at even positions, i.e. d_{2k} for some $k \in \mathbb{N}$, get control bit 0 attached, and data items at odd positions, i.e. d_{2k+1} for some $k \in \mathbb{N}$, get control bit 1 attached.

The BRP is specified in Figure 1. Our specification adapts the nondeterministic process algebra specification of [Fok07] by refining the configuration of lossy channels. While in the nondeterministic setting a lossy channel (nondeterministically) either successfully transmits a datum d_i or loses it, we attached a success and failure probability to this choice. The protocol specification $BRP(N, T, p, q)$ is parametrized by the quadruple (N, T, p, q) , with N denoting the maximum length of the data stream, T denoting how often a single datum may be retransmitted, p the probability that a single attempt to transmit a datum may fail, and q the probability that the acknowledgment may fail. The term $BRP(N, T, p, q)$ represents a system consisting of the RC interface to the TV modeled as process $RC(N, T, p, q)$, the TV interface to the RC modeled as process TV , and the channels $CH(b, t, p, q)$ for data transmission and $CH_2(d, b, t, p, q)$ for acknowledgment.

The processes $RC(N, T, p, q)$ and TV synchronize over the actions:

- (i) $c(d, b)$, with $d \in D$ and $b \in \{0, 1\}$, modeling the correct transmission of datum $d \in D$ and control bit $b \in \{0, 1\}$ from the RC to the TV;
- (ii) ack , modeling the correct transmission of the acknowledgment message from the TV to the RC, and
- (iii) $lost$, used to model the timeout due to loss of the acknowledgment message.

Timeout due to the loss of pair (d, b) is modeled by action \perp by the RC.

The process $RC(N, T, p, q)$ starts by receiving the size $n \leq N$ of the data stream by some other RC component, by means of action $i(n)$. Then, for n times it reads the datum d_i from some other RC components by means of action $i(d)$ and tries to send it to the TV . If all n data are sent successfully, then the other RC components are notified by means of action $res(OK)$. In case of T failures for one datum, the whole transmission fails and the other RC components are notified by means of action $res(NOK)$. If the process TV receives a pair (d, b) from $RC(N, T, p, q)$ by action $c(d, b)$, then, if the datum d is original, namely b is the expected control bit, then d is sent to the other TV components by means of action $o(d)$, otherwise (d, b) is ignored.

To advocate bisimulation metric semantics as a suitable distance measure for performance validation of communication protocols we translate performance properties of a BRP implementation with lossy channels $BRP(N, T, p, q)$ to the bisimulation distance between such an implementation and the specification with perfect channels $BRP(N, T, 0, 0)$. In the following we assume that $\lambda = 1$, namely no discount.

Proposition 5.1. *Let $N, T \in \mathbb{N}$ and $p, q \in [0, 1]$.*

- (1) *The bisimulation distance $\mathbf{d}(BRP(N, T, 0, 0), BRP(N, T, p, q)) = \epsilon$ relates as follows to the protocol performance properties:*
 - (a) *The likelihood that N data items are sent and acknowledged without any retry (this means $BRP(N, T, p, q)$ behaves as $BRP(N, T, 0, 0)$) is $1 - \epsilon$.*
 - (b) *The likelihood that N data items are sent and acknowledged with exactly k retries, for some $0 \leq k \leq N \cdot T$, is $(1 - \epsilon)(1 - (1 - \epsilon)^{1/N})^k$.*
 - (c) *The likelihood that N data items are sent and acknowledged with at most $k \leq N \cdot T$ retries is $(1 - \epsilon) \frac{1 - (1 - (1 - \epsilon)^{1/N})^{k+1}}{(1 - \epsilon)^{1/N}}$.*
 - (d) *The likelihood that at least $n \leq N$ of the N data items are sent and acknowledged is $(1 - \epsilon) \frac{1 - (1 - (1 - \epsilon)^{1/N})^{nT+1}}{(1 - \epsilon)^{1/n}}$.*
 - (e) *The likelihood that all N items are sent and acknowledged is $(1 - \epsilon) \frac{1 - (1 - (1 - \epsilon)^{1/N})^{N \cdot T + 1}}{(1 - \epsilon)^{1/N}}$.*

- (2) *The bisimulation distance $\mathbf{d}(CH(b, T, 0, 0), CH(b, T, p, q)) = \delta$ relates as follows to the channel performance properties:*
- (a) *The likelihood that one datum is sent and acknowledged without any retry is $1 - \delta$.*
 - (b) *The likelihood that one datum is sent and acknowledged with exactly k retries, for some $k \leq T$, is $(1 - \delta) \cdot \delta^k$.*
 - (c) *The likelihood that one datum is sent and acknowledged with at most k retries, for some $k \leq T$, is $1 - \delta^{k+1}$.*
 - (d) *The likelihood that one datum is sent and acknowledged is $1 - \delta^{T+1}$.*

Proof. (1) First we note that $((1 - p)(1 - q))^N$ is the likelihood that N data items are sent and acknowledged without any retry.

- (a) The result can be understood by observing that $\epsilon = 1 - ((1 - p)(1 - q))^N$ is the likelihood that at least one retry is needed to transmit the stream of N data.
 - (b) The result can be understood by observing that $(1 - \epsilon)(1 - (1 - \epsilon)^{1/N})^k$ is the conjunct probability to have exactly k failures in sending or acknowledging a datum (probability $(1 - (1 - \epsilon)^{1/N})^k$), and to have N successes (probability $1 - \epsilon$).
 - (c) The result can be understood by observing that $(1 - \epsilon) \frac{1 - (1 - (1 - \epsilon)^{1/N})^{k+1}}{(1 - \epsilon)^{1/N}} = \sum_{i=0}^k (1 - \epsilon)(1 - (1 - \epsilon)^{1/N})^i$, where $(1 - \epsilon)(1 - (1 - \epsilon)^{1/N})^i$ is the likelihood to send the N data with exactly i retries (see item (b)).
 - (d) This is item (c) with N instantiated with n and k instantiated with $n \cdot T$.
 - (e) This is item (c) with k instantiated with $N \cdot T$.
- (2) First we note that the likelihood that a single datum requires no retry is $(1 - p)(1 - q)$.
- (a) The result can be understood by observing that $\delta = 1 - (1 - p)(1 - q)$ is the likelihood that a single datum requires at least one retry to be successfully transmitted and acknowledged.
 - (b) The result can be understood by observing that $(1 - \delta) \cdot \delta^k = (1 - p)(1 - q) \cdot (1 - (1 - p)(1 - q))^k$ is the conjunct probability to have k failures (probability $(1 - (1 - p)(1 - q))^k$) followed by a successful transmission (probability $(1 - p)(1 - q)$).
 - (c) The result can be understood by observing that $1 - \delta^{k+1} = \sum_{i=0}^k (1 - \delta) \cdot \delta^i$, where $(1 - \delta) \cdot \delta^i$ is the likelihood that one datum is sent and acknowledged with exactly i retries (see item (b)).
 - (d) This is item (c) instantiated with $k = T$.

□

Now we show that by applying the compositionality results given in the previous sections (Propositions 3.1, 3.2, 4.1) we can relate the bisimulation distance between the specification with perfect channels $BRP(N, T, 0, 0)$ and some implementation with lossy channel $BRP(N, T, p, q)$ of the entire protocol with the distances between the specification and some implementation of its respective components. On the one hand, this allows us to derive from specified performance properties of the entire protocol individual performance requirements of its components (compositional verification). On the other hand, this allows us to infer from performance properties of the protocol components suitable performance guarantees on the entire protocol (compositional specification). We show also that the same compositionality results allow us to relate the distance between the specification and some implementation with lossy channel of the entire protocol or some components to the parameters of the system.

Proposition 5.2. *Let $N, T \in \mathbb{N}$ and $p, q \in [0, 1]$. For all $b \in \{0, 1\}$ it holds:*

$$\mathbf{a[ph]^*} \mathbf{d}(BRP(N, T, 0, 0), BRP(N, T, p, q)) \leq 1 - (1 - \mathbf{d}(CH(b, T, 0, 0), CH(b, T, p, q)))^N;$$

$$\mathbf{a[ph]^*} \mathbf{d}(CH(b, T, 0, 0), CH(b, T, p, q)) \leq 1 - (1 - p)(1 - q).$$

$$\mathbf{d}(BRP(N, T, p, q), BRP(N, T, 0, 0)) \leq 1 - ((1 - p)(1 - q))^N$$

Proof. Consider case $\mathbf{a}(ph)^*$. By Proposition 3.2.(d) we obtain $\mathbf{d}(BRP(N, T, 0, 0), BRP(N, T, p, q)) \leq \mathbf{d}(RC(N, T, 0, 0), RC(N, T, p, q)) + (1 - \mathbf{d}(RC(N, T, 0, 0), RC(N, T, p, q)))\mathbf{d}(TV, TV)$. By $\mathbf{d}(TV, TV) = 0$ we get $\mathbf{d}(BRP(N, T, 0, 0), BRP(N, T, p, q)) \leq \mathbf{d}(RC(N, T, 0, 0), RC(N, T, p, q))$. Then, by applying Propositions 3.1.(a), 3.1.(b), 3.2.(a), and 4.1.(a) we infer $\mathbf{d}(RC(N, T, 0, 0), RC(N, T, p, q)) \leq 1 - (1 - \mathbf{d}(CH(b, T, 0, 0), CH(b, T, p, q)))^N$.

Case $\mathbf{a}(ph)^*$ follows directly from Proposition 3.1. More precisely, by Proposition 3.1.(a) we infer both inequalities $\mathbf{d}(CH(b, t, p, q), CH(b, t, 0, 0)) \leq p + (1 - p)\mathbf{d}(CH_2(d, b, t, p, q), CH_2(d, b, t, 0, 0))$ and $\mathbf{d}(CH_2(d, b, t, p, q), CH_2(d, b, t, 0, 0)) \leq q$, which give $\mathbf{d}(CH(b, T, 0, 0), CH(b, T, p, q)) \leq p + (1 - p)q = 1 - (1 - p)(1 - q)$.

Case $\mathbf{a}(ph)^*$ follows directly from cases $\mathbf{a}(ph)^*$ and $\mathbf{a}(ph)^*$. \square

To advocate uniform continuity as adequate property for compositional reasoning, we show that the uniform continuity of process combinators in $BRP(N, T, p, q)$ allows us to relate the distance between this implementation and the specification $BRP(N, T, 0, 0)$ (which relates by Proposition 5.1 to performance properties of the entire protocol) to the concrete parameters p, q and N of the system. In detail, by Theorems 3.5, 3.8, 4.6 we can derive that $\mathbf{d}(BRP(N, T, p, q), BRP(N, T, 0, 0)) \leq N/2 \cdot (\mathbf{d}(CH(0, T, p, q), CH(0, T, 0, 0)) + \mathbf{d}(CH(1, T, p, q), CH(1, T, 0, 0)))$ (see the proof of Proposition 5.3 below). Then, by Proposition 5.2 we can derive $N/2 \cdot (\mathbf{d}(CH(0, T, p, q), CH(0, T, 0, 0)) + \mathbf{d}(CH(1, T, p, q), CH(1, T, 0, 0))) \leq N(1 - (1 - p)(1 - q))$. Summarizing, we can conclude that $\mathbf{d}(BRP(N, T, p, q), BRP(N, T, 0, 0)) \leq N(1 - (1 - p)(1 - q))$, which allows us to infer an upper bound to $\mathbf{d}(BRP(N, T, p, q), BRP(N, T, 0, 0))$ from suitable constraints for p and q , as formalized in the following result.

Proposition 5.3. *Let $N, T \in \mathbb{N}$ and $p, q \in [0, 1]$. For all $\epsilon \geq 0$, $p + q - pq < \epsilon/N$ ensures*

$$\mathbf{d}(BRP(N, T, p, q), BRP(N, T, 0, 0)) < \epsilon$$

Proof. Assume N is even. Then:

$$\begin{aligned} & \mathbf{d}(BRP(N, T, p, q), BRP(N, T, 0, 0)) \\ & \leq \mathbf{d}(RC(N, T, p, q), RC(N, T, 0, 0)) + \mathbf{d}(TV, TV) && \text{(Theorem 3.8)} \\ & = \mathbf{d}(RC(N, T, p, q), RC(N, T, 0, 0)) \\ & \leq \mathbf{d}((CH(0, T, p, q); CH(1, T, p, q))^{N/2}, (CH(0, T, 0, 0); CH(1, T, 0, 0))^{N/2}) && \text{(Theorem 3.5)} \\ & \leq N/2 \cdot \mathbf{d}(CH(0, T, p, q); CH(1, T, p, q), CH(0, T, 0, 0); CH(1, T, 0, 0)) && \text{(Theorem 4.6)} \\ & \leq N/2 \cdot (\mathbf{d}(CH(0, T, p, q), CH(0, T, 0, 0)) + \mathbf{d}(CH(1, T, p, q), CH(1, T, 0, 0))) && \text{(Theorem 3.8)} \\ & = N(1 - (1 - p)(1 - q)) \end{aligned}$$

where in the third inequality we use the Lipschitz factor n for the operator $_{}^n$ that we obtained in the proof of Theorem 4.6. From $\mathbf{d}(BRP(N, T, p, q), BRP(N, T, 0, 0)) \leq N(1 - (1 - p)(1 - q))$ the thesis follows. The case that N is odd is analogous. \square

Combining Propositions 5.1 – 5.3 allows us now to reason compositionally over a concrete scenario. We derive from a given performance requirement to transmit a stream of data the necessary performance properties of the channel components.

Example 5.4. Consider the following scenario. We want to transmit a data stream of $N = 20$ data items with at most $T = 1$ retry per data item. We want to build an implementation that

should satisfy the performance property ‘The likelihood that all 20 data items are successfully transmitted is at least 99%’. By applying Proposition 5.1.1 we translate this performance property to the bisimulation distance $\mathbf{d}(BRP(N, T, 0, 0), BRP(N, T, p, q)) \leq 0.01052$ on the entire system. By applying Proposition 5.2.1^(ph) we derive the bisimulation distance for its channel component $\mathbf{d}(CH(b, T, 0, 0), CH(b, T, p, q)) \leq 0.00053$. By Proposition 5.2.1^(ph) this distance can be translated to appropriate parameters of the channel component, e.g. $p = 0.0002$ and $q = 0.00032$ or equivalently $p = 0.020\%$ and $q = 0.032\%$. Finally, Proposition 5.1.2 allows to translate the distance between the specification and implementation of the channel component back to an appropriate performance requirement, e.g. ‘The likelihood that one datum is successfully transmitted is at least 99.95%’.

6. CONCLUSIONS

We argued that the notion of uniform continuity (Definition 4.4, generalizing the notions of non-expansiveness and non-extensiveness discussed by other researchers) is an appropriate property of process combinators to facilitate compositional reasoning w.r.t. bisimulation metric semantics. We showed that all standard (non-recursive and recursive) process algebra operators are uniformly continuous (Theorems 3.5, 3.8, 4.6, 4.7). In addition, we provided for all standard process algebra operators tight bounds on the distance between the composed processes (Propositions 3.1, 3.2, 4.1). We exemplified how these results can be used to reason compositionally over protocols. In fact, they allow us to derive from performance requirements on the entire system appropriate performance properties of the respective components, and in reverse to induce from performance assumptions on the system components performance guarantees on the entire system.

We remark that the abstraction operator of probabilistic process algebras (that hides actions and makes them observable as non-distinguishable τ -actions) is non-extensive. However, the power of abstraction and hiding can only be utilized by using also a behavioral semantics that treats the τ -actions respectively as internal actions. We leave the development of weak and branching bisimulation metrics and the analysis of process algebra operators for those metrics as future work. A first analysis for weak bisimulation metric and observational congruence weak bisimulation metric (weak bisimulation metric with kernel equivalence being the largest congruence w.r.t. CSS operators contained in weak bisimulation equivalence) may be found in [DJGP02].

The metric reasoning approach exemplified in Section 5 is a sound method to reason compositionally over systems. However, the distance between composed systems might not be tight. Let $C[x]$ be an open term describing a composed system with x the placeholder for a subsystem. Given subsystems s and s' , the distance $\mathbf{d}(C[s], C[s'])$ might be below the composition of the compositionality properties of the operators in C if some of the differences in the behaviors between s and s' do not induce different behaviors between $C[s]$ and $C[s']$. To exemplify this effect, consider the context $C[x] = x \mid b.0$ and subsystems $s = a.0$ and $s' = a.([1 - \epsilon/\lambda]\epsilon \oplus [\epsilon/\lambda]0)$. Clearly $\mathbf{d}(s, s') = \epsilon$. Then the compositional analysis gives $\mathbf{d}(C[s], C[s']) \leq \epsilon$. However, $\mathbf{d}(C[s], C[s']) = 0$ because the behavioral distance between s and s' (observable only after executing action a) cannot be observed in the context $C[x]$ (which can only perform an action if the instances of x perform action b). Thus, $\mathbf{d}(C[s], C[s']) = 0$ since s and s' agree on the inability to perform action b . One idea to tackle this problem is to develop the notion of context bisimulation. Given a context C , the C -bisimulation distance (bisimulation distance w.r.t. context C) between s and s' would measure only that degree of the bisimulation distance between s and s' that would induce different behavior between x instantiated by s and x instantiated by s' . Using the notation \mathbf{d}_C for the C -bisimulation distance this would give the behavioral distance $\mathbf{d}_C(s, s') = 0$ (since C derives only behavior from an initial b -move and s and s' agree on their inability to perform b -moves), while $\mathbf{d}_C(b.0, b.([1 - \epsilon/\lambda]\epsilon \oplus [\epsilon/\lambda]0)) = \epsilon$.

It is clear that the context bisimulation distance is bounded by the bisimulation distance. While it still allows for sound compositional metric reasoning it may lead to tighter bounds. We leave the detailed technical development and analysis as future work.

Another research direction is to generalize the analysis of concrete process algebra operators as discussed in this paper to general SOS rule and specification formats. The basic observation is that the compositionality results for the concrete probabilistic process algebra operators depend only on the specification rules of those operators, hence the question boils down to develop SOS meta-theoretical results and appropriate rule and specification formats that guarantee that the specified operators are uniformly continuous. In essence, we aim to develop the quantitative analogous of the well-established meta-theory for behavioral equivalence semantics [AFV01, MRG07]. This approach has been already developed for notions of approximate probabilistic bisimulation [Tin08, Tin10, GT13]. Preliminary results show that in essence, a process combinator is uniformly continuous if the combined processes are copied only finitely many times along their evolution [GT14, GT15, Geb15], and more restrictive constraints guarantee the stronger compositional properties of Lipschitz continuity, non-expansiveness and non-extensiveness. By following the *divide and congruence* approach [FvGdW06, FvGdW12, GF12, FvG16, CGT16b], formats for compositional properties can be obtained also through a suitable logical characterization of bisimilarity metric, like that in [CGT16a].

Finally, we intend to explore further (as initiated in Section 5) the relation between various behavioral distance measures, e.g. convex bisimulation metric [DAMRS07], trace metric [FL14], and total-variation distance based metrics [Mio14] with performance properties of communication and security protocols. This will provide further practical means to apply process algebraic methods and compositional metric reasoning w.r.t. uniformly continuous process combinators.

REFERENCES

- [ABV94] Luca Aceto, Bard Bloom, and Frits Vaandrager. Turning SOS rules into equations. *Information and Computation*, 111(1):1–52, 1994.
- [AFV01] Luca Aceto, Wan J. Fokkink, and Chris Verhoef. Structural operational semantics. In *Handbook of Process Algebra*, pages 197–292. Elsevier, 2001.
- [And99] Suzana Andova. Process algebra with probabilistic choice. In *Proc. ARTS’99*, volume 1601 of *LNCS*, pages 111–129. Springer, 1999.
- [And02] Suzana Andova. *Probabilistic process algebra*. PhD thesis, Eindhoven University of Technology, 2002.
- [Bar04] Falk Bartels. *On generalised coinduction and probabilistic specification formats*. PhD thesis, VU University Amsterdam, 2004.
- [BBLM13] Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, and Radu Mardare. Computing behavioral distances, compositionally. In *Proc. MFCS’13*, volume 8087 of *LNCS*, pages 74–85. Springer, 2013.
- [BIM95] Bard Bloom, Sorin Istrail, and Albert R. Meyer. Bisimulation can’t be traced. *Journal of ACM*, 42:232–268, 1995.
- [CGPX14] Konstantinos Chatzikokolakis, Daniel Gebler, Catuscia Palamidessi, and Lili Xu. Generalized bisimulation metrics. In *Proc. CONCUR’14*, volume 8704 of *LNCS*, pages 32–46. Springer, 2014.
- [CGT16a] Valentina Castiglioni, Daniel Gebler, and Simone Tini. Logical characterization of bisimulation metrics. In *Proc. QAPL 2016*, EPTCS, 2016.
- [CGT16b] Valentina Castiglioni, Daniel Gebler, and Simone Tini. Modal decomposition on nondeterministic probabilistic processes. In *Proc. CONCUR’16*, volume 59 of *LIPICs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.
- [DAHMO3] Luca De Alfaro, Thomas A. Henzinger, and Rupak Majumdar. Discounting the Future in Systems Theory. In *Proc. ICALP’03*, volume 2719 of *LNCS*, pages 1022–1037. Springer, 2003.
- [DAMRS07] Luca De Alfaro, Rupak Majumdar, Vishwanath Raman, and Mariëlle Stoelinga. Game relations and metrics. In *Proc. LICS’07*, pages 99–108. IEEE, 2007.

- [DCPP06] Yuxin Deng, Tom Chothia, Catuscia Palamidessi, and Jun Pang. Metrics for Action-labelled Quantitative Transition Systems. In *Proc. QAPL'05*, volume 153 of *ENTCS*, pages 79–96, 2006.
- [DD07] Yuxin Deng and Wenjie Du. Probabilistic barbed congruence. In *Proc. QAPL'07*, volume 190 of *ENTCS*, pages 185–203, 2007.
- [DD09] Yuxin Deng and Wenjie Du. The Kantorovich metric in computer science: A brief survey. In *Proc. QAPL'09*, volume 253 of *ENTCS*, pages 73–82, 2009.
- [DGJP04] José Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labelled Markov Processes. *Theoretical Computer Science*, 318(3):323–354, 2004.
- [DGL15] Pedro R. D'Argenio, Daniel Gebler, and Matias D. Lee. A general SOS theory for the specification of probabilistic transition systems. Accepted for I&C. Also available at <http://www.few.vu.nl/~gebler/paper/sos-theory.pdf>, 2015.
- [DJGP02] José Desharnais, Radha Jagadeesan, Vineet Gupta, and Prakash Panangaden. The metric analogue of weak bisimulation for probabilistic processes. In *Proc. LICS'02*, pages 413–422. IEEE, 2002.
- [DL12] Pedro R. D'Argenio and Matias D. Lee. Probabilistic transition system specification: Congruence and full abstraction of bisimulation. In *Proc. FoSSaCS'12*, volume 7213 of *LNCS*, pages 452–466. Springer, 2012.
- [DvGH*07] Yuxin Deng, Rob J. van Glabbeek, Matthew Hennessy, Carroll Morgan, and Chenyi Zhang. Remarks on testing probabilistic processes. In *Computation, Meaning, and Logic: Articles dedicated to Gordon Plotkin*, volume 172 of *ENTCS*, pages 359–397, 2007.
- [FL14] Uli Fahrenberg and Axel Legay. The quantitative linear-time-branching-time spectrum. *Theoretical Computer Science*, 538:54–69, 2014.
- [Fok07] Wan J. Fokkink. *Modelling distributed systems*. Springer, 2007.
- [FvG16] Wan J. Fokkink and Rob J. van Glabbeek. Divide and congruence II: from decomposition of modal formulas to preservation of delay and weak bisimilarity. *CoRR*, abs/1604.07530, 2016.
- [FvGdW06] Wan J. Fokkink, Rob J. van Glabbeek, and Paulien de Wind. Compositionality of Hennessy-Milner logic by structural operational semantics. *Theoretical Computer Science*, 354(3):421–440, 2006.
- [FvGdW12] Wan J. Fokkink, Rob J. van Glabbeek, and Paulien de Wind. Divide and congruence: From decomposition of modal formulas to preservation of branching and η -bisimilarity. *Information and Computation*, 214:59–85, 2012.
- [Geb15] Daniel Gebler. *Robust SOS specifications of probabilistic processes*. PhD thesis, VU University Amsterdam, 2015.
- [GF12] Daniel Gebler and Wan J. Fokkink. Compositionality of probabilistic Hennessy-Milner logic through structural operational semantics. In *Proc. CONCUR 2012*, volume 7454 of *LNCS*, pages 395–409. Springer, 2012.
- [GLT15] Daniel Gebler, Kim G. Larsen, and Simone Tini. Compositional metric reasoning with Probabilistic Process Calculi. In *Proc. FoSSaCS'15*, volume 9034 of *LNCS*, pages 230–245. Springer, 2015.
- [GT13] Daniel Gebler and Simone Tini. Compositionality of approximate bisimulation for probabilistic systems. In *Proc. EXPRESS/SOS'13*, volume 120 of *EPTCS*, pages 32–46, 2013.
- [GT14] Daniel Gebler and Simone Tini. Fixed-point characterization of compositionality properties of probabilistic processes combinators. In *Proc. EXPRESS/SOS'14*, volume 160 of *EPTCS*, pages 63–78, 2014.
- [GT15] Daniel Gebler and Simone Tini. SOS specifications of probabilistic systems by uniformly continuous operators. In *Proc. CONCUR'15*, volume 42 of *LIPICs*, pages 155–168. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.
- [HJ94] Hans Hansson and Bengt Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.
- [JLY01] Bengt Jonsson, Kim G. Larsen, and Wang Yi. Probabilistic extensions of Process Algebras. In *Handbook of Process Algebra*, pages 685–710. Elsevier, 2001.
- [Kel76] Robert M. Keller. Formal verification of parallel programs. *Communications of the ACM*, 19(7):371–384, 1976.
- [LGD12] Matias D. Lee, Daniel Gebler, and Pedro R. D'Argenio. Tree rules in probabilistic transition system specifications with negative and quantitative premises. In *Proc. EXPRESS/SOS'12*, volume 89 of *EPTCS*, pages 115–130, 2012.
- [LS91] Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94:1–28, 1991.
- [LT05] Ruggero Lanotte and Simone Tini. Probabilistic congruence for semistochastic generative processes. In *Proc. FoSSaCS'05*, volume 3441 of *LNCS*, pages 63–78. Springer, 2005.

- [LT09] Ruggero Lanotte and Simone Tini. Probabilistic bisimulation as a congruence. *ACM Transactions on Computational Logic*, 10:1–48, 2009.
- [Mio14] Matteo Mio. Upper-expectation bisimilarity and Łukasiewicz μ -Calculus. In *Proc. FoSSaCS'14*, volume 8412 of *LNCS*, pages 335–350. Springer, 2014.
- [MRG07] Mohammad Reza Mousavi, Michel A. Reniers, and Jan Friso Groote. Sos formats and meta-theory: 20 years after. *Theoretical Computer Science*, 373(3):238–272, 2007.
- [MS13] Matteo Mio and Alex Simpson. A proof system for compositional verification of probabilistic concurrent processes. In *Proc. FoSSaCS'13*, volume 7794 of *LNCS*, pages 161–176. Springer, 2013.
- [Pan09] Prakash Panangaden. *Labelled Markov Processes*. Imperial College Press, 2009.
- [Seg95] Roberto Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, MIT, 1995.
- [SL95] Roberto Segala and Nancy Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2:250–273, 1995.
- [Ste94] William J. Stewart. *Introduction to the numerical solution of Markov Chains*. Princeton University Press, 1994.
- [Tin08] Simone Tini. Non expansive ϵ -bisimulations. In *Proc. AMAST'08*, volume 5140 of *LNCS*, pages 362–376. Springer, 2008.
- [Tin10] Simone Tini. Non-expansive ϵ -bisimulations for probabilistic processes. *Theoretical Computer Science*, 411:2202–2222, 2010.
- [vB12] Franck van Breugel. On behavioural pseudometrics and closure ordinals. *Information Processing Letters*, 112(19):715–718, 2012.
- [vBW01] Franck van Breugel and James Worrell. Towards quantitative verification of probabilistic transition systems. In *Proc. ICALP'01*, volume 2076 of *LNCS*, pages 421–432. Springer, 2001.
- [vBW05] Franck van Breugel and James Worrell. A behavioural pseudometric for probabilistic transition systems. *Theoretical Computer Science*, 331(1):115–142, 2005.
- [Vil08] Cédric Villani. *Optimal transport: old and new*. Springer, 2008.