VRIJE UNIVERSITEIT

Robust SOS Specifications of Probabilistic Processes

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. V. Subramaniam,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Exacte Wetenschappen
op vrijdag 13 november 2015 om 9.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

Ekkehart Daniel Gebler

geboren te Räkelwitz, Duitsland

promotor:	prof.dr. W.J. Fokkink
copromotor:	dr. P.R. D'Argenio

Robust SOS Specifications of Probabilistic Processes

Daniel Gebler

IPA Dissertation Series 2015-20
ISBN: 978-90-5383-169-4
A catalogue record is available from the VU University Amsterdam Library

The work in this thesis has been carried out under the auspices of the research school IPA (Institute for Programming research and Algorithmics). The author was employed at the VU University Amsterdam.

Contents

Preface

Doing a PhD is a wonderful journey of constant challenges. First, I would like to thank my supervisors Wan Fokkink and Pedro R. D'Argenio for their continued support and the great scientific freedom they provided. While it became quickly clear that the initial research question would be only an incremental research contribution, my desire to work on more fundamental research questions and to explore significant and profound ideas grew. The recently developed metric behavioral semantics was not only untapped territory for the language specification community but meant also that we had to fully understand and combine numerous non-trivial concepts from pure mathematics such as measure theory and real analysis, and concepts from applied mathematics such as optimization, linear programming and transportation theory. Your uninterrupted support allowed me to explore ideas and directions that may have been initially more peripheral to your core scientific fields. Thank you for the ongoing trust that these ideas will eventually lead to a fully satisfactory consistent and coherent theory. I'm also grateful for the opportunity to run the local research seminar and reading group where we had over the last years many interesting talks and scientific debates.

One of my first and most prolific research contacts was Simone Tini. After an initial contact in early 2012 we started later that year to explore in-depth the connection between metric compositionality properties and language specification properties. Owing to favorable circumstances Simone was looking for a new research direction and had significant amount of research time available. Additionally, we had a very compatible working schedule. This meant in practice frequent Skype calls late at night and substantial research progress also along the weekends. I'm indebted and very thankful to this wonderful prolific collaboration over the last years.

Another fundamental research collaboration started at FoSSaCS'12 in Tallinn (Estonia) where I met Pedro D'Argenio and Matias Lee. After I studied their paper on probabilistic SOS languages I got very much excited about this research direction. The talk itself may have left more questions than answers but it triggered enough interest and curiosity to decide that language specification theory of probabilistic systems should become the main topic of my thesis. Shortly afterwards we published already our first common paper at EXPRESS/SOS'12 and collaborated until recently on numerous papers together. Thanks for the great collaboration over those years, the wonderful time we had in Córdoba including BBQs, unparalleled Argentinian steaks, Mate cocktails and much more. Pedro's group is a great example how a highly committed and dedicated research group can make with very limited funding both significant and lasting contributions.

I attended a number of great research schools that did not only provide a broad per-

spective on formal methods and quantitative systems theory but allowed me also to build up many valuable and lasting connections into the research community. Most prominently let me mention ROCKS'12 in Varna (Italy) where we post-published three survey papers, QMC'12 in Copenhagen (Denmark) which provided a deep understanding of the various notions of probabilistic parallel composition, GAMES'13 in Champéry (Switzerland) which revealed a surprisingly simple connection between operational and logical semantics in terms of bisimulation games, SMC'14 in Lyon (France) which made the importance of real analysis and topology for quantitative specification and verification theory clear, and MOVEP'14 in Nantes (France) which provided extensive feedback on my line of research and future scientific ambitions.

A very important contribution to the notions of compositionality developed in this thesis is due to Kim G. Larsen and the Schloss Dagstuhl seminar on "Quantitative Models: Expressiveness, Analysis, and New Applications" (January 2014, Germany). While discussing with Kim existing notions of metric compositionality he outlined his proposal of uniform continuity. Even if the notion of uniform continuity could not be applied (at this time) to compute behavioral distances compositionally, it should become the right notion to specify languages compositionally. Even more, it should become the unifying notion of metric compositionality and the foundation for the language specification theory developed in this thesis. A research visit of the Kim's group in Aalborg (Denmark) in spring 2015 and extensive discussions also with Radu, Giovanni, and Giorgio proved very helpful to understand deeply the properties of bisimulation metric semantics. I'm very grateful to Kim for the Dagstuhl invitation and the productive fruitful follow-up collaboration.

Another great research experience started at CONCUR'13 in Buenos Aires. After meeting with Marco Bernardo and discussing the status of the concurrency community it got clear that we needed to reactivate collaboration and stimulate the search for new substantial profound ideas in concurrency theory. We organized together with Michele Loreti the invitation only workshop "Open Problems in Concurrency Theory" at the wonderful medieval town of Bertinoro (Italy). The healthy mix between talks on recent research results, survey like overviews, and controversial talks on emerging unhealthy trends and situations provided a great platform for productive conversations. The special issue journal of this workshop will ensure a lasting effect to the concurrency research community.

Aside organizing workshops and conferences, it was also a great pleasure to serve as a PC member in 2014 and PC chair in 2015 and 2016 for the EXPRESS/SOS workshop. While the PC membership may have been a logical consequence of being quite active in the community, the PC chairing was an unexpected honor. Chairing the workshop in 2015 (where my co-chair Silvia Crafa couldn't attend), running the steering committee meeting, kicking-off multiple new initiatives and actively headhunting for a co-chair for 2016 was a great learning experience! I'm thankful to Silvia Crafa and Johannes Borgström for offering me this opportunity.

I got numerous invitations to present my work and discuss recent research results. I'm very grateful for all these opportunities which certainly helped me to form and finalize this thesis. Without claiming to be exhaustive I would like to thank Holger Hermanns, Tom Henzinger, Joost-Pieter Katoen, Kim Larsen, Radu Mardare, Rocco De Nicola, Catuscia Palamidessi, and Daniele Varacca. Appologies to all those where I couldn't follow the invitation. Additionally I would like to thank Luca Aceto, Yuxin Deng, Matteo Mio, Mohammad Mousavi for valuable discussions and comments over the years.

Bringing the research of multiple years together in a thesis is not only demanding for the author but means also that the thesis committee needs to take time of their busy schedules to read the dissertation. I would like to thank Matthew Hennessy, Joost-Pieter Katoen, Jan Willem Klop, and Mohammad Mousavi for reading the thesis carefully and provide valuable feedback to improve the manuscript.

Performing theoretical research means that a small community meets regularly on exotic places to discuss topics which may seem obscure to the external observer. Nevertheless, this strengthen the community inside and many friendships arise naturally. Special thanks goes to Alexandra, Christian, David, Dennis, Florian, Helle, Jurriaan, Lili, Gijs, Mark, Matteo, Michele, Giorgio, Giovanni, Hassan, Simone, and Vahid. Similarly, also my colleagues at the VU provided a great environment to pursue research, esp. Andrew, Bas, Clemens, David, Dimitri, Femke, Jan Willem, Jeroen, Jörg, Rena, Roel, Stefan, and Wan.

The academic research trajectory has been complemented by a wonderful business and startup experience which I wouldn't have wanted to miss. First and foremost to mention is my recent involvement in the online supermarket startup Picnic. Big thanks to Frederik and Joris for their patience, support and trust that this PhD journey takes a happy end. Thanks also to the exceptionally gifted technology enthusiasts Alecio, Bruno, Eva, Fernando, Henrique, Keith, Iliana, Jasper, Joris-Jan, Lars, Marc, Niek, Paul, Sid, Siebe, Stephan, Sunny, Victoria (and soon Renske) that make this entrepreneurial environment so exciting. Thanks also to all the growth hackers and business veterans in Picnic which will without any doubt make this ambition a major success! Thanks also to my former Fredhopper colleagues and friends Andreas, David & Nynke, Francisca & Stan, Frank, Georg & Christy, Jens & Lietsa, Lucas, Nikolay, Peter & Wendy, Ronny & Rocio, and Tim.

Last but not least such a long research trajectory would not have been possible without the continued support of my family. The dense travel schedule and demanding deadlines did challenge Stephanie and my kids Julian and Annika at times but I hope that the souvenirs and growing Lego-collection could make up for it. Thank you for your love and support!

Daniel Gebler, 2015

Chapter 1

Introduction

Information and computing systems play a fundamental role in nearly all aspects of the modern society. They are central in many industries such as transportation, energy, finance, healthcare, and telecommunication, as well as ubiquitous in almost all aspects of daily life. Many of those systems are safety-critical whose malfunctioning may result in significant harm on people, environment, or property. The increasing complexity of those systems requires rigorous modeling and analysis techniques to specify, verify, and validate their requirements and behavior. Formal methods are popular mathematically based techniques used in computing science that provide the necessary means to support the development process from design to deployment, to ensure correct behavior.

For instance, consider a navigation satellite system consisting of earth orbiting satellites that are deployed in satellite constellations to provide geo-spatial positioning services [HLW08]. Many systems in transportation, aviation, and defense depend on the correct functioning of the navigation satellite system. Incorrect or missing position data, miscommunication of data, and many more malfunctions could result in loss of life, significant property damage, or damage to the environment. Hence navigation satellite systems are one of the prime examples of safety- and mission-critical systems.

In this context formal methods are applied as follows. First, we specify the system by describing in mathematical terms the system requirements and expectations of the system user. Then we formulate the concrete functional behavior and non-functional properties of the actual systems. Finally we verify that the actual system satisfies precisely (or approximately) the system requirements. While on the one hand precise satisfaction of functional properties is essential, on the other hand approximate or partial satisfaction of non-functional properties may be enough. For instance, it may suffice that a system satisfies the performance requirements by only an acceptance interval of $\pm 5\%$.

An essential component of the navigation satellite system is its communication protocol. It allows for communication and coordination of the satellites and for the terrestrial remote management of the satellite system. The communication protocol is formalized by describing the sending unit, the receiving unit and the communication logic between sender and receiver. In the remainder of the introduction we will refer to this sketched communication protocol in order to explain the fundamental concepts and approaches of formal specification and verification.

1.1 Research context

The advent of large-scale distributed and concurrent systems requires formal specification and verification methods which capture both *qualitative* and *quantitative* properties of systems [HS06]. The scientific context of this thesis is defined by an appropriate formal model, an adequate formal semantics and a suitable formal language to describe and reason over those systems.

Operational model

We start by discussing a few important models that capture qualitative and quantitative aspects of the system behavior separately and later introduce a unified model that represents both aspects together and has a suitable structure for compositional reasoning. Here we emphasize on intuitive explanations rather than mathematical rigor.

Labeled transition systems [Kel76], LTSs for short, are directed graph structures that capture the qualitative (functional) aspects of software and hardware systems. An LTS is an event-based model that describes the operational reactive behavior of a discrete system in terms of states and action-labelled transitions. Formally, a system model consists of a set of states S and transitions of the form $s \xrightarrow{a} s'$ (with $s, s' \in S$) describing that the system in state s can evolve to state s' by performing a transition with label a. To exemplify the LTS model, consider again the communication protocol sketched above. There are various states of the LTS that represent, respectively, the sender and receiver in different configurations. For instance, there is one state that represents the sender in the event of having sent a single data item, and there is another state that represents the receiver in the event of not having received that data item yet. The evolution of the system is modeled by transitions between the states. Action labels allow us to describe the kind of computation or interaction steps performed by system. A typical action label of a communication protocol would describe that a data item gets sent, or that a data item is received. The synchronized execution of both actions describes the successful transmission of data formalized as interaction between the sender and receiver.

LTSs allow us to model uncertainty about the reactive behavior of a system by nondeterministic choices. Nondeterminism is essential to model scheduling freedom, implementation freedom, abstractions or uncertainties in the external environment, and incomplete information [Hoa85; Alf97; Seg95; Sto02]. For instance, the choice between different implementations of some behavior such as the initiation of a transmission in a communication protocol may be a nondeterministic choice. Technically, this is modeled by a state with multiple equally labelled outgoing transitions. Those transitions describe the possible reactions of a system to the same event. Concrete implementations of the system will resolve those nondeterministic choices by schedulers that select one of the possible choices.

Quantitative (or random) phenomena occur whenever the behavior of a system is not deterministic and the uncertainty can be quantified. They arise in nearly every system either by construction or from the physical properties of the system and its environment. Probability is one of the most important measures of uncertainty and has become indispensable in areas such as networks, data mining, security, artificial intelligence, embedded systems, bioinformatics and many more. For instance, distributed algorithms like the

Zeroconf protocol [SC05] are by construction based on random choices to break symmetry. Similarly, cryptographic protocols like SSL insert uncertainty in order to achieve their goals. Additionally, the physical properties of transmission channels determine that message passing between distributed systems suffers typically from a failure rate. The corruption or loss of messages on a channel may be the result of unreliability of the transmission medium (system internal property) or of interference and collisions with other concurrent transmissions (system external environment property).

Quantitative phenomena are typically modeled by Markov chains [KS60]. The most prominent probabilistic model is the Discrete Time Markov Chain [Ste94; HJ94], DTMC for short. In DTMCs the time evolves in discrete steps, i.e., the system performs one operation per clock tick, and the evolution of the system is determined only by the current state and the specified probabilistic choices (Markov property). Formally, a DTMC model consists of a set of states S and transitions of the form $s \longrightarrow \pi$, with π a distribution over S, describing that the system in state s can evolve to state s' with probability $\pi(s')$. DTMCs are widely used to analyze and evaluate the performance of computer and communication systems [Hav01]. An important extension of DTMCs are probabilistic LTSs [LS91; GSS95] that decorate probabilistic transitions with action labels. On the one hand, action labels allow us to express the kind of computation that a system executes. On the other hand, they provide also the means to express various notions of communication, synchronization and coordination between systems in terms of appropriate composition operators. Formally, a probabilistic LTS model is an action-indexed family of DTMCs that describe for each action label a by transition $s \longrightarrow_a \pi$ that the system in state s can evolve to state s' with probability $\pi(s')$ by performing action a. However, probabilistic LTSs do not allow for nondeterministic choices, i.e., for each action label the successor states are determined only by the probabilistic choices. To summarize, LTSs are powerful models to specify and verify the functional correctness of systems, while DTMCs and probabilistic LTSs are suitable models to evaluate the performance of systems. However, as argued in [Seg95; HS06] and above, both nondeterminism and probability are essential to model distributed concurrent and embedded systems.

Nondeterministic probabilistic labeled transition systems, PTSs for short, combine LTSs and DTMCs by separately modeling the reactive system behavior, nondeterministic choices and probabilistic choices. PTSs have been introduced as probabilistic automata in [Seg95] and are successfully applied to model distributed and concurrent systems [Seg95; Sto02]. Essentially, a system is modeled by a set of states S and a set of transitions of the form $s \xrightarrow{a} \pi$ that describe that the system in state s may perform a transition labelled with a and evolve to a state s' with probability $\pi(s')$. PTSs allow for nondeterministic choices, i.e., there may be different transitions $s \xrightarrow{a} \pi$ and $s \xrightarrow{a} \pi'$ representing that the system in state s may perform action a and then choses nondeterministically either distribution π or π'. We will employ PTSs as the operational model to study the compositionality properties of distributed and concurrent systems.

Behavioral semantics

The development of an appropriate model of a real system is a non-trivial task. A distributed and concurrent system may be represented by different PTS models, e.g., probabilistic choices may be differently encoded, multiple equivalent implementation options

may be given, or naming may be different. Behavioral semantics gives a formal notion to compare systems. Behavioral equivalences are behavioral semantics that allow us to determine the observational equivalence of systems by abstracting from behavior that may not be observable or otherwise not relevant in the application context. The most prominent notion is bisimulation equivalence which dates back to Park and Milner [Mil80; Par81] and provides a well-understood standard framework to express behavioral equivalences of dynamic systems. Two states are bisimilar if they can mimic each other's behavior. There is a plethora of behavioral equivalences that form a linear-time branching-time spectrum [Gla93; Gla90]. In this thesis we will focus on bisimulation semantics due to its canonicity. Nevertheless, we will indicate how the developed techniques and results may be transferred to other behavioral semantics.

The concept of bisimulation equivalence on LTSs was lifted by Larsen and Skou to probabilistic LTSs [LS91], and later by Segala and Lynch to PTSs [SL95; Seg95]. Two states of a PTS are bisimilar if each transition can be mimicked by an equally labelled transition and the probability mass of bisimilar states coincide. This basic idea dates back to the well-known notion of lumpability in Markov chain theory [KS60]. Bisimulation equivalence on PTSs enjoys similar nice properties as bisimulation equivalence on LTSs and allows for neat coinductive, fixed point and logical characterizations [PS07; Her+11]. Similar to the spectrum of behavioral equivalences on LTSs there is also some recent work to develop comparative probabilistic semantics and a spectrum of behavioral equivalences on PTSs [CR11; BDL13].

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 how far the behavior of these systems is apart.

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. There are multiple approaches that aim for a quantitative notion of behavioral semantics. The most prominent notion is behavioral metric semantics [Des+04; BW05; Den+05] which provides a behavioral distance that characterizes how far the behavior of two systems is apart. For instance, behavioral metrics allow us to express that an ideal transmission channel (no transmission errors) is closer to a lossy channel with failure rate ϵ than to another lossy channel with failure rate 2ϵ. 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 pseudometric with bisimilar processes at distance 0. Alternative approaches towards a quantitative semantics for probabilistic processes are approximate bisimulation [GJS90; DLT08; TDZ11; GT13] and bisimulation degrees [Yin02a; Yin02b; ZZ08]. However, aside from the less elegant mathematical formulations of approximate bisimulation and bisimulation degrees, only (discounted) bisimulation metric satisfies important and natural properties that are required to measure the quality of approximation in a meaningful sense (tech-

nically these are parameter and property continuity [Mar+14]). Hence, we follow the convincing argumentation[1] in [GJS90; Des+04; BW05] and will consider in this thesis bisimulation metric semantics.

Compositional specification

Process algebras [Mil89; Hoa85; Hen88; Fok13] are languages to formally describe the behavior of concurrent systems in a compositional way. A process algebra provides a set of primitive operators (also called process combinators or process connectors) that allow us to describe complex systems by successively composing subsystems and primitive components. Probabilistic process algebras [JLY01; Den15] are process algebras with probabilistic connectors that allow us to describe the behavior of distributed concurrent systems. The probabilistic connector describes a probabilistic choice that is the result of some uncertain behavior which cannot be affected and does not interact or depend on the environment. The probabilistic choice provides information on the outcome of the choice but abstracts away from the details of how the choice is made. Each process term of a probabilistic process algebra induces a PTS that describes the operational semantics of the respective system.

The operational semantics of process algebras and programming languages is usually described by Structural Operational Semantics (SOS) specifications [Plo81]. An SOS specification assigns to each language expression a transition system with transitions inductively defined by means of SOS rules. The terms are inductively defined over the alphabet of the language. The rules define how a process should behave (i.e., perform certain activities) in terms of the behavior of its subprocesses. In other words, the rules define compositionally the transition system associated to each term of the language. Intuitively, the terms denote programs or processes with an associated stepwise computation behavior described by the state transition relation.

An important early insight [Sim84; Sim85] in SOS research was that many properties of process algebra and programming language operators depend only on the structure of the SOS inference rules used to specify their behavior. This started the important branch of SOS meta-theory that investigates which syntactic properties of the rules ensure by construction semantic properties of the induced transition systems [GV92; BIM95; Gro93; BG96; MRG07]. The main objective of this research branch is to define constraints on the structure of the rules and specifications (called rule and specification formats) that guarantee, by construction, a given semantic property.

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

[1] The importance of behavioral metric semantics has also been recognized by international funding agencies, e.g., ERC advanced grant "Quantitative Reactive Modeling" [Hen13], 2010, Tom Henzinger, and ERC advanced grant "Learning, Analysis, SynthesiS and Optimization of Cyber-Physical Systems", 2015, Kim G. Larsen.

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 as it allows to reason about systems in an equational framework. Also 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.

On these grounds, the congruence property is one of the most studied properties of SOS rules and specifications. Over the last decades, numerous rule and specification formats have been defined that ensure the congruence property on the induced LTS w.r.t. various behavioral equivalences, see e.g. [AFV01b; MRG07] for an overview. One of the most important formats is the GSOS format [BIM95] that ensures congruence of the specified operators w.r.t. bisimulation equivalence on LTS. A first attempt to generalize this format to the probabilistic setting was made in [Bar02; LT05; LT09] by using literals of the form $t \xrightarrow{a,p} t'$ that decorate transitions with a probability in order to partially specify a probabilistic jump. However, this approach required complicated consistency conditions to ensure that all partially specified probabilistic jumps together define a valid transition to a full distribution. Hence, Bartels and later D'Argenio and Lee [Bar04; DL12] introduced the probabilistic GSOS format that uses triples of the form $t \xrightarrow{a} \theta$ (with t a state term and θ a distribution term) to specify in a single literal all probabilistic choices of a transition. The probabilistic GSOS format [Bar04; DL12; LGD12; DGL14; DGL15] ensures that bisimulation equivalence is a congruence w.r.t. all operators specified by rules in this format. Finally we remark that we introduced in [DL12; LGD12] the even more expressive $nt\mu f\theta/nt\mu x\theta$ format which allows additionally for look-ahead and still ensures that bisimulation equivalence is a congruence for the specified operators.[2] Nevertheless, look-ahead is not compatible with metric compositionality and allows to define operators that are not compositional w.r.t. bisimulation metric semantics [GT13]. Hence we employ throughout this thesis the probabilistic GSOS format.

1.2 Research questions

The PTS operational model, bisimulation metric semantics and probabilistic GSOS specifications provide the formal context for the research questions explored in this thesis. While the theory of compositional specification and reasoning w.r.t. behavioral equivalence semantics is reasonably well-developed [Bai+04; RS14; DGL15; Den15], there exists so far no adequate theory in the context of behavioral metric semantics. The objective and goal of this thesis is to develop a specification theory of probabilistic nondeterministic systems w.r.t. bisimulation metric semantics. This provides language designers, system developers, and operations engineers with the necessary theory, methods and tools to specify and verify those systems in an efficient compositional way.

[2] Additionally we proposed in [DLG15a] sub-formats of the $nt\mu f\theta/nt\mu x\theta$ format for convex bisimulation equivalence, probability obliterated bisimulation equivalence, and probability abstracted bisimulation equivalence. Moreover, Lee and de Vink proposed in [LV15] a sub-format for rooted branching bisimulation equivalence.

Compositional metric reasoning

A language allows for compositional reasoning if the behavioral semantics is in some sense compatible with all operators of the language. While for behavioral equivalence semantics compositionality is given by the well-understood and extensively studied property of congruence, 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 parameter are modified in a controlled manner). Earlier proposals such as non-expansiveness [Des+04] and non-extensiveness [Bac+13] 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. This leads us to the first research questions:

What is an appropriate notion of metric compositionality? (Q1)

We study this question in Chapter 3 and argue that uniform continuity (generalizing non-expansiveness and non-extensiveness) captures the essential nature of compositional metric reasoning and allows us now to reason also compositionally about recursive processes. 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. Technically, this is expressed by a modulus of continuity that relates the behavioral distance between the parts s and s' with the behavioral distance between the composed systems $C[s]$ and $C[s']$. 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 investigate this newly proposed notion of metric compositionality in the context of probabilistic process calculi and analyze which operators are uniformly continuous and hence allow for compositional reasoning w.r.t. bisimulation metric semantics. Uniform continuity does not only formalize when an operator is compositional but quantifies also by means of the modulus of continuity how the distance between composed processes relates to the distance between their subprocesses. Hence we derive also for all standard operators of probabilistic process calculi the least possible modulus of continuity, i.e. a tight bound on the distance between composed processes given the distance between their parts. Combining these results, we demonstrate how compositional metric reasoning

about systems specified by uniformly continuous process algebra operators allows for metric assume-guarantee like performance validation.

Specification of compositional operators

With an appropriate notion of metric compositional reasoning at hand and after having gained understanding of how this notion applies to some concrete probabilistic process algebra, the natural next question is how those results can be generalized to arbitrary process algebras and programming languages. By observing that the compositionality results for the concrete probabilistic process algebra operators depended only on the specification rules of those operators, the question boils down to developing SOS meta-theoretical results and appropriate rule and specification formats that guarantee that the specified operators are uniformly continuous. While there is a rich and well established SOS metatheory for behavioral equivalence semantics [AFV01b; MRG07], the following pressing question has been open until now:

How can we specify compositional process combinators? (Q2)

In Chapter 4 we investigate which SOS specifications define uniformly continuous operators. Aside the canonical metric compositionality property of uniform continuity we consider also the stricter properties of Lipschitz-continuity, non-expansiveness, and non-extensiveness that allow to specify bounded recursive operators, non-recursive operators, and behavioral choice operators. The stricter compositionality properties play an important role to get a tight bound on the distance between the terms built of operators with different compositionality properties. Building on that analysis, we develop (a spectrum of) SOS rule and specification formats that allow us to simultaneously specify operators with different compositionality properties (e.g., one operator is non-extensive and another operator is Lipschitz continuous) in one SOS specification.

Conceptually, the specification of compositional operators works as follows. A language designer defines for each operator the required compositionality property in terms of the modulus of continuity. The specification format determines then structural properties of the specification rules to ensure that a) the kernel relation of the bisimulation metric (i.e. bisimulation equivalence) is a congruence relation w.r.t. all operators of the specified language, and b) the language operators satisfy the defined modulus of continuity. We obtain property a) by defining our format as a subformat of the probabilistic GSOS format (which satisfies already the congruence property), and achieve property b) by determining carefully the additional rule and specification restrictions to satisfy the required modulus of continuity. Each rule and specification format exploits the (possibly different) compositionality guarantees of all operators used in the specification rules. This admits an expressive class of specifications.

Our formats allow us to derive the modulus of continuity of process algebra and programming language operators by simple inspection of the syntactic pattern of the specification rules. We remark that our specification format is the first that allows to specify simultaneously operators of different compositionality properties. We show how the modulus

of continuity can be used to derive a bound on the distance between two closed instances of a partial program specification or open process algebra expressions. More general, our SOS meta-theoretical results are the very first for bisimulation metric semantics. The only earlier attempt to answer the question Q2 was done by Tini in [Tin08; Tin10] by considering only probabilistic LTSs (i.e. excluding non-deterministic choice), employing approximate bisimulation equivalence (i.e. a notion of bisimulation distance with major drawbacks), and by analyzing only non-expansiveness as compositionality property.

A denotational model of metric compositionality

In order to answer the first two questions we developed tools and methods to specify and reason about probabilistic nondeterministic systems. However, those methods and tools provide only partial and indirect insights into the fundamental relation between structural specification properties and the related metric compositionality properties of the specified operators. Hence, in order to get a deeper understanding of the relation between syntactical properties and the associated metric behavioral semantics properties, the following question arises:

Which primitive process behavior determines the compositionality property of a process combinator?	(Q3)

In Chapter 5 we analyze the class of all finite probabilistic processes (specified by basic probabilistic process algebra operators) and deduce that process replication, probabilistic choice, and nondeterministic choice are the primitive operational process behavior that determine the moduli of continuity of the specified operators. This allows us to define a denotational model that formalizes these primitive behaviors in a stratified manner. The domain model allows us to systematically analyze how the various primitive process behavior interact, e.g. that a unary operator that replicates its argument once but evolves only by probability 0.5 has the same modulus of continuity as a unary operator that does not replicate its argument but evolves by probability of 1. Reversely, the domain model allows us to analyze for a given modulus of continuity (understood as compositionality requirement) what is the optimal set of primitive operational process behavior that satisfies this modulus of continuity.

The denotational model associates with each operator a denotation that characterizes the primitive operational behavior of processes combined by that operator. The denotation of an operator is computed by recursively counting how many times the combined processes are replicated along their evolution, weighted by the likelihood of the replication. The modulus of continuity of an operator can then directly be derived from its denotation. More fundamentally, we will show that compositionality on the syntactic level (composing operators to terms), semantic level (composing moduli of continuity of operators to gain moduli of continuity of terms) and denotational level (composing denotations of operators to gain denotations of terms) coincide.

The denotational model opens the door for a new approach to derive SOS rule and specification formats. Traditionally, SOS rule formats were developed by an *explorat-*

ive approach as follows: First, one analyzes the operational and compositional behavior induced by the structural properties of a representative set of rules. Then one generalizes (and simplifies) the structural properties of all those rules that specify operators of the desired semantical property. As a final step one defines the rule format by (reasonable) argumentation for admissible rules satisfying the generalized structural properties, typically by showing that obvious relaxations of the format constraints may violate the intended semantical property. Prominent examples are the GSOS format [BIM95] and the *ntyft/ntyxt* [Gro93] format, as well as more recently the probabilistic GSOS format [DL12; LGD12] and the newly introduced metric GSOS format of Chapter 4. An alternative approach was enabled by the development of compositional logical proof systems for the satisfaction relation of HML-formulae [LX91; FGW06a; GF12]. The *logical approach* derives SOS rule formats from the logical characterization of the behavioral equivalence under investigation [BFG04; FGW06b; FGW06c; FGW12].

We propose a new *denotational approach* that derives an adequate SOS rule and specification format from the denotational model that defines the required compositionality properties of the operators. One of the interesting benefits of this approach is that it allows us to derive for a given modulus of continuity different rule formats from which the language designer may select the best one based on his application context.

Axiomatizing bisimulation equivalences and metrics

At this point we have gained a comprehensive understanding of compositional metric reasoning, the required underlying formal notions and developed a powerful toolbox that allows us to specify new probabilistic process algebras and programming languages. In other words, we have developed SOS meta-theoretical results that relate the syntactic properties of specification languages with the operational and metric semantics behavior of the specified operators. However, to understand algebraic properties of the specified operators, such as associativity, commutativity, idempotence, distributivity etc., the perspective of algebraic semantics and equational reasoning is required. Hence, we round up the thesis by raising the following question:

Which equational algebraic properties do compositional process combinators satisfy?	(Q4)

In chapter 6 we develop a method that generates from any probabilistic GSOS specification a sound and ground-complete equational axiomatization for bisimulation equivalence. The construction is based on the nondeterministic GSOS axiomatization method of [ABV94] and the equational theory for probabilistic languages of [BS01; Hen12]. The novelty in our approach is to employ multi-sorted algebras to axiomatize separately nondeterministic choice, probabilistic choice and its interaction. Furthermore, we generalize this method to the axiomatize metric bisimulation distance of probabilistic GSOS specifications.

This method allows us to analyze the algebraic properties of operators and processes. For instance, it allows to derive in a systematic way by formal equational reasoning that

parallel composition is commutative and associative, i.e. $x \mid y = y \mid x$ and $(x \mid y) \mid z = x \mid (y \mid z)$. In a similar way, it allows to derive that parallel composition is non-expansive, i.e. $d(x_1 \mid x_2, y_1 \mid y_2) \leq d(x_1, y_1) + d(x_2, y_2)$, and that the n-time iteration of processes is n-Lipschitz, i.e. $d(x^n, y^n) \leq n \cdot d(x, y)$. Hence, it provides an alternative approach to derive the modulus of continuity of operators. For instance, we can derive that an operator f specified as $f(x) = x \mid x$ is 2-Lipschitz continuous.

1.3 Organization of the thesis

The remainder of the thesis is organized as follows. The preliminaries in Chapter 2 cover all standard technical definitions and important known results of the research context described above. If the reader is familiar with probabilistic concurrency theory and structural operational semantics, then he may directly dive into the technical material of Chapters 3–6. Otherwise, a basic study of the preliminaries is advisable to allow for a gentle study and easy reading of the technical material. Each of the technical Chapters 3–6 answers one of the research questions Q1 – Q4. When studying the technical material for the first time it may be easiest if the reader focuses first on the formal concepts, their intuitive motivation, the clarification by examples, and the key results. Since the research context and results are very technical by nature, we opted to provide all proofs in the main text to make it easy for the interested reader to study also immediately the argumentation and reasoning that lead to the results. Each chapter closes with a section that briefly summarizes the results and outlines proposals for possible future research directions. In a few cases we sketch possible solution ideas for the raised future research questions that may be used to continue the line of research of this thesis. The thesis concludes in Chapter 7 by summarizing the research questions and results and providing a comprehensive outlook to possible future research directions.

The thesis covers the main results of the published papers [GLT15; GT13; GT14; GT15; DGL14]. We developed a number of other results that are published in [GF12; LGD12; GGM13; Cha+14; DGL15; DLG15a; Yux15] but are not covered in the thesis. Besides that we would like to refer the interested reader to our survey papers [Deh+14; GHT14; Arn+14] which provide an extensive introduction to probabilistic modeling and verification and offer many references to standard literature and recent advances in the field.

Chapter 2

Preliminaries

This chapter provides the necessary mathematical background used in the thesis. We limit ourselves to those notions and concepts that will be used later in the technical chapters. An introduction to the standard concepts from lattice theory, probability theory, and set theory may be found in [DP02; Fel08; Kal06; Hal60].

We start by introducing algebraic languages in Section 2.1, then define the operational model in Section 2.2, followed by the behavioral semantics in Section 2.3, and finish with the specification framework in Section 2.4. Most of the material is standard except for the important new result in Theorem 2.33 and Corollary 2.34 that shows how moduli of continuity of language operators distribute over probabilistic choices. This result will be of fundamental importance for the development of the specification formats.

2.1 Algebraic languages

A formal language is defined by an alphabet and a grammar. We will use algebraic languages that are defined by a signature and a simple grammar given by the freely generated term algebra that defines the sentences as terms over the alphabet.

Definition 2.1 (Signature)**.** A *signature* is a structure $\Sigma = (F, \mathsf{r})$, where F is a countable set of *operators*, and $\mathsf{r} \colon F \to \mathbb{N}$ is a *rank function*.

The rank function gives by $\mathsf{r}(f)$ 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 $\mathsf{r}(f)$. We may write $f \in \Sigma$ as shorthand for $\Sigma = (F, \mathsf{r})$ 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 $\mathsf{T}(\Sigma, V)$, is the least set satisfying:

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

We write c for $c()$ if c is a constant. The set of *closed state terms* $\mathsf{T}(\Sigma, \emptyset)$ is abbreviated as $\mathsf{T}(\Sigma)$. The set of *open state terms* $\mathsf{T}(\Sigma, \mathcal{V}_s)$ is abbreviated as $\mathbb{T}(\Sigma)$. To clarify the notation of operands in terms, we may write operators together with underscores to denote the position of the operands. For instance, for a binary operator f, we may write $f\_\_$ when using the prefix notation, $\_\_f$ when using the postfix notation, and $\_f\_$ when using the infix notation. In the context of process algebra, we may refer to operators as *process combinators*, to state variables as *process variables*, and to closed state terms as *processes*.

Notation 2.3 (Notations for state terms)**.** Let $t \in \mathbb{T}(\Sigma)$ be any term. We denote by $\mathsf{Var}(t)$ the set of all state variables in t, i.e. $\mathsf{Var}(x) = \{x\}$ and $\mathsf{Var}(f(t_1, \ldots, t_n)) = \bigcup_{i=1}^{n} \mathsf{Var}(t_i)$. If the number of occurrences of variables is important, we will use the expression $\mathsf{Var}(t, x)$ to denote how many times the variable x occurs in term t, defined by $\mathsf{Var}(x, x) = 1$, $\mathsf{Var}(x, y) = 0$ if $x \neq y$, and $\mathsf{Var}(f(t_1, \ldots, t_n), x) = \sum_{i=1}^{n} \mathsf{Var}(t_i, x)$. By $\mathsf{depth}(t)$, we mean the *depth* of term t, defined by $\mathsf{depth}(x) = 0$ and $\mathsf{depth}(f(t_1, \ldots, t_n)) = 1 + \max_{i=1}^{n} \mathsf{depth}(t_i)$.

Probability distributions formalize the concept of probabilistic choices by describing how likely a specific state is selected. Technically, a probability distribution is a mapping $\pi \colon \mathsf{T}(\Sigma) \to [0, 1]$ with $\sum_{t \in \mathsf{T}(\Sigma)} \pi(t) = 1$ that assigns to each closed term $t \in \mathsf{T}(\Sigma)$ its respective probability $\pi(t)$. We denote by $\Delta(\mathsf{T}(\Sigma))$ the set of all probability distributions on $\mathsf{T}(\Sigma)$. We let π, π' range over $\Delta(\mathsf{T}(\Sigma))$.

Notation 2.4 (Notations for probability distributions)**.** The probability mass of a set of closed terms $T \subseteq \mathsf{T}(\Sigma)$ in some probability distribution $\pi \in \Delta(\mathsf{T}(\Sigma))$ is given by $\pi(T) = \sum_{t \in T} \pi(t)$. We denote by $\delta(t)$ with $t \in \mathsf{T}(\Sigma)$ the *Dirac distribution* defined by $(\delta(t))(t) = 1$ and $(\delta(t))(t') = 0$ if $t \neq t'$. The convex combination $\sum_{i \in I} p_i \pi_i$ of a family $\{\pi_i\}_{i \in I}$ of probability distributions $\pi_i \in \Delta(\mathsf{T}(\Sigma))$ 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 $t \in \mathsf{T}(\Sigma)$. The expression $f(\pi_1, \ldots, \pi_n)$ with $f \in \Sigma$ and $\pi_i \in \Delta(\mathsf{T}(\Sigma))$ denotes the product distribution of $\pi_1, \ldots, \pi_n$ defined by $(f(\pi_1, \ldots, \pi_n))(f(t_1, \ldots, t_n)) = \prod_{i=1}^{n} \pi_i(t_i)$ and $(f(\pi_1, \ldots, \pi_n))(t) = 0$ for all terms $t \in \mathsf{T}(\Sigma)$ not in the form $t = f(t_1, \ldots, t_n)$. 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 2.5 (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 $\mathsf{DT}(\Sigma, V_s, V_d)$, is the least set satisfying:

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

Distribution terms have the following meaning. A *distribution variable* $\mu \in \mathcal{V}_d$ is a variable that takes values from $\Delta(\mathsf{T}(\Sigma))$. 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* $\mathsf{DT}(\Sigma, \emptyset, \emptyset)$ is abbreviated as $\mathsf{DT}(\Sigma)$. The set of *open distribution terms* $\mathsf{DT}(\Sigma, \mathcal{V}_s, \mathcal{V}_d)$ 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, we may write $\theta_1 \, f \, \theta_2$ for $f(\theta_1, \theta_2)$.

Notation 2.6 (Notations for distribution terms)**.** For a distribution term $\theta \in \mathbb{DT}(\Sigma)$, we denote by $\mathsf{Var}(\theta)$ the set of all state and distribution variables in θ, i.e. $\mathsf{Var}(\mu) = \{\mu\}$, $\mathsf{Var}(\delta(t)) = \mathsf{Var}(t)$, $\mathsf{Var}(\sum_{i \in I} p_i \theta_i) = \bigcup_{i \in I} \mathsf{Var}(\theta_i)$, and $\mathsf{Var}(f(\theta_1, \ldots, \theta_n)) = \bigcup_{i=1}^{n} \mathsf{Var}(\theta_i)$. Then, we denote by $\mathsf{Var}(\theta, \zeta)$ the number of occurrences of variable ζ in term θ, i.e. $\mathsf{Var}(\mu, \mu) = 1$, $\mathsf{Var}(\mu, \nu) = 0$ if $\mu \neq \nu$, $\mathsf{Var}(\delta(t), x) = \mathsf{Var}(t, x)$, $\mathsf{Var}(\sum_{i \in I} p_i \theta_i, \zeta) = \sum_{i \in I} p_i \mathsf{Var}(\theta_i, \zeta)$, and $\mathsf{Var}(f(\theta_1, \ldots, \theta_n), \zeta) = \sum_{i=1}^{n} \mathsf{Var}(\theta_i, \zeta)$. By $\mathsf{depth}(\theta)$ we mean the depth of distribution term θ defined by $\mathsf{depth}(x) = 0$, $\mathsf{depth}(\delta(t)) = \mathsf{depth}(t)$, $\mathsf{depth}(\sum_{i \in I} p_i \theta_i) = \max_{i \in I} \mathsf{depth}(\theta_i)$, and $\mathsf{depth}(f(\theta_1, \ldots, \theta_n)) = 1 + \max_{i=1}^{n} \mathsf{depth}(\theta_i)$.

Definition 2.7 (Substitution)**.** A *substitution* is a mapping $\sigma \colon \mathcal{V} \to \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)$ 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 extends to a mapping from state terms to state terms by $\sigma(f(t_1, \ldots, t_n)) = f(\sigma(t_1), \ldots, \sigma(t_n))$. A substitution extends to a mapping from distribution terms to distribution terms by $\sigma(\delta(t)) = \delta(\sigma(t))$, $\sigma(\sum_{i \in I} p_i \theta_i) = \sum_{i \in I} p_i \sigma(\theta_i)$, and $\sigma(f(\theta_1, \ldots, \theta_n)) = f(\sigma(\theta_1), \ldots, \sigma(\theta_n))$.

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

In order to simplify later the proofs and argumentation over distribution terms, we will introduce a normal form of distribution terms. A distribution term $\theta \in \mathbb{DT}(\Sigma)$ is in normal form if either there is no convex combination in θ or there is only one convex combination as outermost operation. Moreover, the Dirac operator is only applied to single variables $\delta(x)$ and not to non-variable terms $\delta(t)$ with $t \notin \mathcal{V}_s$.

Definition 2.8 (Distribution terms in normal form)**.** A distribution term $\theta \in \mathbb{DT}(\Sigma)$ is in *normal form* iff either

- $\theta = \mu \in \mathcal{V}_d$, or
- $\theta = \delta(x)$ with $x \in \mathcal{V}_s$, or
- $\theta = \sum_{i \in I} p_i \theta_i$ where all θ_i are in normal form and no convex combination appears in any of the θ_i, or
- $\theta = f(\theta_1, \ldots, \theta_n)$ where all θ_i are in normal form and no convex combination appears in any of the θ_i.

We call distribution terms $\theta_1, \theta_2 \in \mathbb{DT}(\Sigma)$ equivalent, notation $\theta_1 \equiv \theta_2$, if $\sigma(\theta_1) = \sigma(\theta_2)$ for all closed substitutions σ. In other words, $\theta_1 \equiv \theta_2$ iff the closed instances $\sigma(\theta_1)$ and $\sigma(\theta_2)$ denote the same distribution. For each distribution term an equivalent distribution term in normal form can be constructed.

Proposition 2.9. *For each $\theta_1 \in \mathbb{DT}(\Sigma)$ there is a $\theta_2 \in \mathbb{DT}(\Sigma)$ in normal form with $\theta_1 \equiv \theta_2$.*

Proof. We construct θ_2 from θ_1 by recursively applying

$$\begin{aligned}
\delta(f(t_1,\ldots,t_n)) &= f(\delta(t_1),\ldots,\delta(t_n)) \\
\sum_{i=1}^{n} p_i \sum_{j=1}^{n_i} p_{i,j}\theta_{i,j} &= \sum_{\substack{i=1\ldots n \\ j=1\ldots n_i}} (p_i \cdot p_{i,j})\theta_{i,j} \\
f\left(\sum_{j=1}^{n_1} p_{1,j}\theta_{1,j},\ldots,\sum_{j=1}^{n_n} p_{n,j}\theta_{n,j}\right) &= \sum_{\substack{i=1\ldots n \\ j_i=1\ldots n_i}} \prod_{i=1}^{n} p_{i,j_i} f(\theta_{1,j_1},\ldots,\theta_{n,j_n}).
\end{aligned}$$

Equivalence $\theta_1 \equiv \theta_2$ follows by soundness of the equations. It is easy to see that the resulting term θ_2 is in normal form. □

2.2 Probabilistic transition systems

Since the main objective of our study is formal languages, we will define all concepts, structures and notations specific to this language context. For instance, we will define the transition systems not w.r.t. arbitrary state spaces but w.r.t. the language expressions for which we want to specify the operational semantics.

Probabilistic nondeterministic labelled transition systems [Seg95], PTSs for short, extend labelled transition systems [Arn94] by allowing for probabilistic choices in the transitions. As state space we will take the set of all closed terms $\mathsf{T}(\Sigma)$. The transitions will be specified later inductively by means of so-called SOS rules.

We define now PTSs as transition systems over the state space of all closed state terms and labelled transitions from terms to distributions over terms.

Definition 2.10 (PTS, [Seg95])**.** A *probabilistic nondeterministic labeled transition system* (*PTS*) over the signature Σ is given by a triple $(\mathsf{T}(\Sigma), A, \longrightarrow)$, where

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

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

Example 2.11. We start by defining a simple language. Let $\Sigma = (\{\_ \mid \_, s, t, 0\}, \mathsf{r})$ be a signature with $\mathsf{r}(|) = 2$ and $\mathsf{r}(s) = \mathsf{r}(t) = \mathsf{r}(0) = 0$. In other words, $s, t, 0$ are constants and $|$ is a binary operator (introduced in detail later in Chapter 3 as synchronous parallel

composition operator). Then, $s \mid t$ is a closed state term and $x \mid x$, with x a state variable, is an open state term. The term variables are $\mathsf{Var}(s \mid t) = \emptyset$ and $\mathsf{Var}(s \mid t, x) = 0$ for all $x \in \mathcal{V}_s$, and $\mathsf{Var}(x \mid x) = \{x\}$, $\mathsf{Var}(x \mid x, x) = 2$ and $\mathsf{Var}(x \mid x, y) = 0$ if $x \neq y$. Moreover, $\mathsf{depth}(x) = 0$, $\mathsf{depth}(s) = \mathsf{depth}(t) = \mathsf{depth}(0) = \mathsf{depth}(x \mid x) = 1$ and $\mathsf{depth}(s \mid t) = 2$.

Now we define a PTS over the language defined above. Let $(\mathsf{T}(\Sigma), A, \rightarrow)$ be a PTS over Σ with actions $A = \{a\}$ and transitions $\rightarrow = \{s \xrightarrow{a} \delta(s), t \xrightarrow{a} 0.5\delta(t) + 0.5\delta(0), s \mid t \xrightarrow{a} 0.5\delta(s \mid t) + 0.5\delta(s \mid 0)\}$. Note that $0.5\delta(t) + 0.5\delta(0)$ is a distribution with $(0.5\delta(t) + 0.5\delta(0))(t) = (0.5\delta(t) + 0.5\delta(0))(0) = 0.5$ and $(0.5\delta(t) + 0.5\delta(0))(t') = 0$ for any term $t' \in \mathsf{T}(\Sigma)$ with $t \neq t' \neq 0$.

2.3 Bisimulation semantics

Behavioral semantics are comparative semantics that assign a meaning to PTSs by defining appropriate notions of behavioral equivalence or behavioral distance between states.

2.3.1 Bisimulation equivalence

Behavioral equivalences are equivalence relations that relate states which cannot be distinguished by an external observer. There are various notions of external observer in terms of different discrimination behavior, e.g. based on branching trees, linear traces and tests [Gla93; Gla90]. One of the most prominent notions for PTSs is bisimulation equivalence [LS91; Seg95] that relates states which mimic each other's behavior in a step-wise manner.

The classical notion of bisimulation equivalence on labelled transition systems [Mil80; Par81] relates two states if each transition of one state can be mimicked by an equally labelled transition of the other state such that both transitions lead again to related states. In order to adapt this concept to PTSs we need a notion to express that an equivalence relation on states is lifted to an equivalence relation on distributions. This will allow us to express that related states can mimic each other's behavior an lead to related distributions.

The lifting of a state relation to a distribution relation is formalized as follows. A *matching*[1] 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 2.12 (Relational lifting matching based)**.** Let $\mathsf{R} \subseteq \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma)$ be any binary relation. The *lifting* of R is a relation $\overline{\mathsf{R}} \subseteq \Delta(\mathsf{T}(\Sigma)) \times \Delta(\mathsf{T}(\Sigma))$ defined by

$$\pi \,\overline{\mathsf{R}}\, \pi' \quad \text{iff} \quad \exists \omega \in \Omega(\pi, \pi').\, (\forall t, t' \in \mathsf{T}(\Sigma).\, (\omega(t, t') > 0 \Rightarrow t \;\mathsf{R}\; t'))$$

[1] Matchings have been introduced in [JL91; Seg95] under the name weight function with the additional condition that $\omega(t, t')$ implies $t \;\mathsf{R}\; t'$ for all $t, t' \in \mathsf{T}(\Sigma)$. The additional condition is part of the lifting Definition 2.12.

for all $\pi, \pi' \in \Delta(\mathsf{T}(\Sigma))$.

This formalization of lifting is elegant since it allows later for a natural quantitative generalization to behavioral metrics. An equivalent alternative definition [DD11] relates those distributions that have the same probability mass w.r.t. sets closed under the state relation. To formalize this idea, we need the following notation. Let $\mathsf{R} \subseteq \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma)$ be a symmetric relation. We define $\mathsf{R}(X) = \{t' \in \mathsf{T}(\Sigma) \mid \exists t \in X \,.\, t \;\mathsf{R}\; t'\}$ for any set of closed terms $X \subseteq \mathsf{T}(\Sigma)$. We call X *closed* w.r.t. R, notation R-closed(X), if $\mathsf{R}(X) \subseteq X$.

Definition 2.13 (Relational lifting closed set based)**.** Let $\mathsf{R} \subseteq \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma)$ be a symmetric relation. The *lifting* of R is a relation $\overline{\mathsf{R}} \subseteq \Delta(\mathsf{T}(\Sigma)) \times \Delta(\mathsf{T}(\Sigma))$ defined by

$$\pi \,\overline{\mathsf{R}}\, \pi' \quad \text{iff} \quad \pi(X) = \pi'(X) \text{ for all } X \subseteq \mathsf{T}(\Sigma) \text{ with R-closed}(X)$$

for all $\pi, \pi' \in \Delta(\mathsf{T}(\Sigma))$.

Note that for equivalence relations the set of R-closed sets is the set of all equivalence classes closed under union. Hence, for an equivalence relation R the definition simplifies to $\pi \,\overline{\mathsf{R}}\, \pi'$ iff $\pi(X) = \pi'(X)$ for all equivalence classes $X \in \mathsf{T}(\Sigma)/\mathsf{R}$. The lifting allows us now to define the notion of bisimulation equivalence on PTSs.

Definition 2.14 (Bisimulation equivalence)**.** A symmetric relation $\mathsf{R} \subseteq \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma)$ is a *bisimulation equivalence* if whenever $t \;\mathsf{R}\; t'$ and $t \xrightarrow{a} \pi$ then there exists a transition $t' \xrightarrow{a} \pi'$ for a distribution $\pi' \in \Delta(\mathsf{T}(\Sigma))$ such that $\pi \,\overline{\mathsf{R}}\, \pi'$.

We call two terms *bisimilar* if they are related by some bisimulation. Bisimulation equivalences are closed under arbitrary union [Seg95]. Hence, we can define the greatest bisimulation equivalence as the union of all bisimulation equivalences. We call the greatest bisimulation equivalence *bisimilarity equivalence* and denote it by the symbol $\sim$.

Example 2.15. We reconsider the language defined in Example 2.11. Let $(\mathsf{T}(\Sigma), A, \rightarrow)$ be 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 $\pi_t = 0.25\delta(t) + 0.25\delta(s) + 0.5\delta(0)$. To show that $s \sim t$, we need to show that $\pi_s \,\overline{\sim}\, \pi_t$. Assume that $\sim = \{(0,0),(s,s),(t,t),(s,t),(t,s)\}$.

First we consider the lifting based on Definition 2.12. Let $\omega \in \Delta(\mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma))$ be defined by $\omega(s,s) = 0.25$, $\omega(s,t) = 0.25$, $\omega(0,0) = 0.5$. Clearly, $\omega \in \Omega(\pi_s, \pi_t)$ since the left marginal of ω is π_s and the right marginal is π_t. Since ω assigns only those pairs of states a non-zero probability that are related by $\sim$, we conclude $\pi_s \,\overline{\sim}\, \pi_t$.

Alternatively, consider the lifting based on Definition 2.13. The sets $\emptyset, \{0\}, \{s,t\}, \{0,s,t\}$ are all $\sim$-closed sets. Note that $\{s\}$ is not $\sim$-closed since $\sim(\{s\}) = \{s,t\} \not\subseteq \{s\}$ (and similar for $\{t\}$). It is easy to verify that $\pi_s(X) = \pi_t(X)$ for all $\sim$-closed sets X. Hence, $\pi_s \,\overline{\sim}\, \pi_t$.

To summarize: For states s and t with $s \sim t$ we verified that the a-labelled transition $s \xrightarrow{a} \pi_s$ can be mimicked by the equally labelled transition $t \xrightarrow{a} \pi_t$ with $\pi_s \,\overline{\sim}\, \pi_t$. The argument for $t \sim s$ is symmetric. The remaining relations of $\sim$ are reflexive, which is trivial to verify. Hence, $\sim$ is indeed a bisimulation relation. Moreover, it is the largest bisimulation equivalence, i.e. no more states can be related. Hence $\sim$ is also the bisimilarity equivalence.

2.3.2 Bisimulation metric

Bisimulation equivalence is a very elegant and well-developed behavioral equivalence. However, as argued already in the introduction, bisimulation equivalence (and in fact most behavioral equivalences) is too sensitive to the exact probabilities of transitions. The slightest perturbation of the probabilities can destroy bisimilarity.

Bisimulation metric[2] [Des+04; BW05; Den+05] 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 2.16 (Pseudometric)**.** A function $d\colon \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \to [0,1]$ is a *1-bounded pseudometric* if

- $d(t,t) = 0$ for all $t \in \mathsf{T}(\Sigma)$,
- $d(t,t') = d(t',t)$ for all $t, t' \in \mathsf{T}(\Sigma)$ (symmetry), and
- $d(t,t') \leq d(t,t'') + d(t'',t')$ for all $t, t', t'' \in \mathsf{T}(\Sigma)$ (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 (employed in Chapters 3 and 6) and in terms of fixed points (employed in Chapters 4 and 5).

Both characterizations require the following lattice structure. Let $([0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)}, \sqsubseteq)$ be the complete lattice of functions $d\colon \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \to [0,1]$ 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 infinimum 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 $t, t' \in \mathsf{T}(\Sigma)$.

Metrical lifting

Bisimulation metric is characterized using the classical bisimulation game where related states can mimic each other's transitions and evolve to distributions that are again related. In the metric context this means that states 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. Hence, bisimulation metric assigns distances to pairs of states that do not increase along their evolution. Technically, this means that we need, just as in the bisimulation equivalence case, a notion that lifts pseudometrics from states to distributions (to capture probabilistic choices). With this concept at hand, it will be straightforward to define the quantitative notion of bisimulation equivalence.

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

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 analogue to the relational lifting given in Definition 2.12 [BW01].

Definition 2.17 (Kantorovich lifting)**.** Let $d\colon \mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)\to[0,1]$ be a 1-bounded pseudometric. The *Kantorovich lifting* of d is a 1-bounded pseudometric $\mathbf{K}(d)\colon \Delta(\mathsf{T}(\Sigma))\times\Delta(\mathsf{T}(\Sigma))\to[0,1]$ 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.

Example 2.18. We reconsider the language defined in Example 2.11. Assume a 1-bounded pseudometric d with $d(s,s)=d(0,0)=0$ and $d(s,0)=d(0,s)=1$. Let $\pi_s=0.5\delta(s)+0.5\delta(0)$ and $\pi_t=(0.5+\epsilon)\delta(s)+(0.5-\epsilon)\delta(0)$ for some arbitrary $\epsilon\in[0,0.5]$. Let us first consider the matching $\omega_1\in\Omega(\pi_s,\pi_t)$ defined by $\omega_1(s,0)=0.5-\epsilon$, $\omega_1(s,s)=\epsilon$, $\omega_1(0,s)=0.5$. Now, $\sum_{t,t'\in\mathsf{T}(\Sigma)} d(t,t')\cdot\omega_1(t,t')=1-\epsilon$. Another matching is $\omega_2\in\Omega(\pi_s,\pi_t)$ defined by $\omega_2(s,s)=0.5$, $\omega_2(0,s)=\epsilon$, $\omega_2(0,0)=0.5-\epsilon$. Now, $\sum_{t,t'\in\mathsf{T}(\Sigma)} d(t,t')\cdot\omega_2(t,t')=\epsilon$. Even more, ω_2 is the optimal matching, i.e. $\sum_{t,t'\in\mathsf{T}(\Sigma)} d(t,t')\cdot\omega_2(t,t')\le\sum_{t,t'\in\mathsf{T}(\Sigma)} d(t,t')\cdot\omega(t,t')$ for any $\omega\in\Omega(\pi_s,\pi_t)$. Hence, $\mathbf{K}(d)(\pi_s,\pi_t)=\epsilon$.

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

Definition 2.19 (Hausdorff lifting)**.** Let $\hat{d}\colon \Delta(\mathsf{T}(\Sigma))\times\Delta(\mathsf{T}(\Sigma))\to[0,1]$ be a 1-bounded pseudometric. The *Hausdorff lifting* of $\hat{d}$ is a 1-bounded pseudometric $\mathbf{H}(\hat{d})\colon P(\Delta(\mathsf{T}(\Sigma)))\times P(\Delta(\mathsf{T}(\Sigma)))\to[0,1]$ 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 $\hat{d}$.

Example 2.20. We consider again the language defined in Example 2.11, and the pseudometric d and distributions π_s,π_t from Example 2.18. Additionally, let $\pi'=(0.5+\epsilon')\delta(s)+(0.5-\epsilon')\delta(0)$ with $\epsilon'\in[0,0.5]$. The distance between the distributions is $\mathbf{K}(d)(\pi_s,\pi_t)=\epsilon$, $\mathbf{K}(d)(\pi_s,\pi')=\epsilon'$, and $\mathbf{K}(d)(\pi_t,\pi')=|\epsilon-\epsilon'|$. Consider now the sets of distribution $\{\pi_s\}$ and $\{\pi_t,\pi'\}$. Intuitively, the Hausdorff distance between $\{\pi_s\}$ and $\{\pi_t,\pi'\}$ models the bisimulation game, i.e. the game that π_s has to match with either π_t or π', and vice versa. Formally, the distance between $\{\pi_s\}$ and $\{\pi_t,\pi'\}$ is $\mathbf{H}(\mathbf{K}(d))(\{\pi_s\},\{\pi_t,\pi'\})=\max(\inf(\mathbf{K}(d)(\pi_s,\pi_t),\mathbf{K}(d)(\pi_s,\pi')),\sup(\mathbf{K}(d)(\pi_t,\pi_s),\mathbf{K}(d)(\pi',\pi_s)))=\max(\epsilon,\epsilon')$. Yet another example is the distance between $\{\pi_s,\pi'\}$ and $\{\pi_t\}$ given by $\mathbf{H}(\mathbf{K}(d))(\{\pi_s,\pi'\},\{\pi_t\})=\max(\sup(\mathbf{K}(d)(\pi_s,\pi_t),\mathbf{K}(d)(\pi',\pi_t)),\inf(\mathbf{K}(d)(\pi_t,\pi_s),\mathbf{K}(d)(\pi_t,\pi')))=\epsilon$.

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 [AHM03; Des+04]. 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 2.21 (Bisimulation metric [Des+04])**.** A 1-bounded pseudometric $d : \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \to [0,1]$ 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* 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.

Example 2.22. We reconsider the language defined in Example 2.11. Let $(\mathsf{T}(\Sigma), A, \longrightarrow)$ be a PTS with transitions $\longrightarrow = \{s \xrightarrow{a} \pi_s, t \xrightarrow{a} \pi_t\}$ 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 $d(s,0) = d(0,s) = 1$. As shown in Example 2.18 we have $\mathbf{K}(d)(\pi_s,\pi_t) = \epsilon$. 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 $\mathbf{d}(t,s) = \lambda\epsilon$.

Fixed point characterization

We provide now an alternative characterization of bisimulation metric in terms of prefixed points of an appropriate monotone bisimulation functional [Den+05]. 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 steps.

Definition 2.23 (Bisimulation metric functional)**.** Let $\mathbf{B} : [0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)} \to [0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)}$ be the function defined by

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

for $d : \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \to [0,1]$ 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 2.24 ([Den+05])**.** *Let $d : \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \to [0,1]$ be a 1-bounded pseudometric. Then $\mathbf{B}(d) \sqsubseteq d$ iff d is a bisimulation metric.*

Proposition 2.24 provides the fixed point characterization of bisimulation metrics and shows that it coincides with the coinductive characterization of Definition 2.21. 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 2.25 ([Den+05])**.** *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 steps. Intuitively, this is a bisimulation distance based on an external observer that discriminates states by considering only the first k transitions. The resulting notion of up-to-k bisimilarity metric will be a very important tool to prove the compositionality results by induction over the transition structure.

Definition 2.26 (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. Since $\mathbf{B}$ is continuous, the closure ordinal of $\mathbf{B}$ is ω [Bre12]. Hence, up-to-k bisimulation distances converge to the bisimulation distances when $k \to \infty$. This opens the door to show properties of the bisimulation metric by using a simple inductive argument.

Proposition 2.27 ([Bre12])**.** $\mathbf{d} = \lim_{k\to\infty} \mathbf{d}_k$.

Example 2.28. We reconsider the language defined in Example 2.11. Let $(\mathsf{T}(\Sigma), A, \longrightarrow)$ be a PTS with transitions $\longrightarrow = \{s \xrightarrow{a} \pi_s, t \xrightarrow{a} \pi_t\}$ whereby $\pi_s = 0.5\delta(s) + 0.5\delta(0)$ and $\pi_t = (0.5+\epsilon)\delta(t)+(0.5-\epsilon)\delta(0)$ for some arbitrary $\epsilon \in [0, 0.5]$. Note that the distribution π_t has now state t in the support instead of state s as in Example 2.22.

By definition, $\mathbf{d}_0(s,t) = 0$. Hence, $\mathbf{d}_1(s,t) = \lambda \cdot \mathbf{K}(\mathbf{d}_0)(\pi_s, \pi_t) = 0$. However, $\mathbf{d}_1(s,0) = 1$ since $der(s,a) = \{\pi_s\}$, $der(0,a) = \emptyset$ and $\inf\emptyset = 1$. Similarly, we have $\mathbf{d}_1(t,0) = 1$. Thus, $\mathbf{K}(\mathbf{d}_1)(\pi_s, \pi_0) = \epsilon$ and $\mathbf{d}_2(s,t) = \lambda\epsilon$.

The up-to-2 bisimulation distance between π_s and π_t is $\mathbf{K}(\mathbf{d}_2)(\pi_s, \pi_t) = \epsilon + 0.5\lambda\epsilon$. Hence, $\mathbf{d}_3(s,t) = \lambda(\epsilon + 0.5\lambda\epsilon)$. In general, the up-to-$k$ bisimulation distance between s and t is $\mathbf{d}_k(s,t) = \epsilon \sum_{n=0}^{k-2} 0.5^n \lambda^{n+1}$. Hence, the bisimulation distance between states s and t is $\mathbf{d}(s,t) = \lim_{k\to\infty} \mathbf{d}_k(s,t) = \lim_{k\to\infty} \epsilon \sum_{n=0}^{k-2} 0.5^n \lambda^{n+1} = \epsilon \sum_{n=0}^{\infty} 0.5^n \lambda^{n+1} = \epsilon/(1-0.5\lambda)$.

Finally, we remark that bisimulation metrics have been also characterized by real-valued modal logics [Des+04], and in terms of coalgebras [BW05].

Properties of bisimulation metrics

We will discuss now a few important properties of bisimulation metrics that are essential for the argumentation later in the technical chapters. We start by showing that bisimilarity equivalence is the kernel of the λ-bisimilarity metric.

Proposition 2.29 ([Des+04])**.** *Let* $t, t' \in \mathcal{T}(\Sigma)$ *be any terms. Then* $\mathbf{d}(t,t') = 0$ *iff* $t \sim t'$*.*

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 2.30. *Let $d\colon \mathcal{T}(\Sigma)\times\mathcal{T}(\Sigma)\to[0,1]$ be a 1-bounded pseudometric. Then*

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

Proof. We start with Proposition 2.30.1 and reason as follows.

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

Now we show Proposition 2.30.2. By Proposition 2.24 we get that $d(t,t') < 1$ implies $\mathbf{B}(d)(t,t') < 1$. The thesis follows now from Proposition 2.30.1. □

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

Remark 2.31. The bisimulation distance $\mathbf{d}(t,t')$ between terms t and t' is in $[0,\lambda]\cup\{1\}$. If $\lambda\in(0,1)$, then $\mathbf{d}(t,t')=1$ iff t can perform an action which t' cannot (or vice versa), $\mathbf{d}(t,t')=0$ iff t and t' have the same reactive behavior (are bisimilar), and $\mathbf{d}(t,t')\in(0,\lambda]$ iff t and t' have different reactive behavior after performing the same initial actions. If $\lambda=1$ then $\mathbf{d}(t,t')=0$ 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.

Properties of the Kantorovich lifting

The Kantorovich pseudometric satisfies important properties that will be essential to define the specification formats. In detail, the Kantorovich lifting functional is monotone, the Dirac operator is a non-expansive embedding of the metric space of states into the metric space of distributions[3], and probabilistic choice distributes over the Kantorovich lifting.

Proposition 2.32 ([Pan09])**.** *Let d and d' be any 1-bounded pseudometrics. Then*

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

[3] In fact, the Kantorovich lifting operator is an isometric embedding of the metric space of states into the metric space of distributions. However, for the specification formats developed later non-expansiveness suffices.

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. This property is essential to reason compositionally over probabilistic systems. Moreover, Proposition 2.32, Theorem 2.33 and Corollary 2.34 together form the cornerstones for the specification formats developed later in this thesis.

Theorem 2.33. *Let $d\colon T(\Sigma)\times T(\Sigma)\to[0,1]$ be any 1-bounded pseudometric. Assume an n-ary operator $f\in\Sigma$ and a concave*[4] *function $z\colon[0,1]^n\to[0,1]$ with*

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

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

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

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

Proof. We assume $\omega_i\in\Omega(\pi_i,\pi'_i)$ to be an optimal matching such that $\mathbf{K}(d)(\pi_i,\pi'_i)=\sum_{t,t'\in\mathsf{T}(\Sigma)}d(t,t')\cdot\omega_i(t,t')$, 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 $\omega\in\Delta(\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma))$ by

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

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

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

[4] A function $z\colon[0,1]^n\to[0,1]$ is called concave if, for any $x_1,\ldots,x_n,y_1,\ldots,y_n\in[0,1]$ and any $\lambda\in[0,1]$, $z((1-\lambda)x_1+\lambda y_1,\ldots,(1-\lambda)x_n+\lambda y_n)\geq(1-\lambda)z(x_1,\ldots,x_n)+\lambda z(y_1,\ldots,y_n)$.

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

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

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

The proof obligation can be derived now by

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

$$= z\left(\left(\sum_{t_1,t_1'\in\mathsf{T}(\Sigma)} d(t_1,t_1')\omega_1(t_1,t_1'),\ldots,\sum_{t_n,t_n'\in\mathsf{T}(\Sigma)} d(t_n,t_n')\omega_n(t_n,t_n')\right)\right)$$
$$= z(\mathbf{K}(d)(\pi_1,\pi_1'),\ldots,\mathbf{K}(d)(\pi_n,\pi_n'))$$

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

Since linear functions are concave, we get that the Kantorovich lifting preserves linear moduli of continuity of language operators.

Corollary 2.34. *Let $d\colon T(\Sigma)\times T(\Sigma)\to[0,1]$ be any 1-bounded pseudometric. Assume an n-ary operator $f\in\Sigma$ and $K\in\mathbb{R}_{\geq 0}$ with*

$$d(f(t_1,\ldots,t_n),f(t_1',\ldots,t_n')) \leq K\sum_{i=1}^n d(t_i,t_i')$$

for all terms $t_1,t_1',\ldots,t_n,t_n'\in T(\Sigma)$. Then we have

$$\mathbf{K}(d)(f(\pi_1,\ldots,\pi_n),f(\pi_1',\ldots,\pi_n')) \leq K\sum_{i=1}^n \mathbf{K}(d)(\pi_i,\pi_i')$$

for all probability distributions $\pi_1,\pi_1',\ldots,\pi_n,\pi_n'\in\Delta(T(\Sigma))$.

However, non-concave moduli of continuity may not be preserved, e.g. the q-norm with $q>1$ (cf. Example 4.2 on page 56 below).

2.4 PGSOS specifications

Programming languages are algebraic languages with an operational semantics given by an appropriate PTS. The state space of the PTS is formed by the terms of the language (cf. Definitions 2.10). The transitions are usually described by means of Structural Operational Semantics (SOS) specifications. An SOS specification consists of SOS rules which are syntax-driven inference rules that define inductively the behavior of complex language expressions in terms of the behavior of their components [MRG07]. An important class of programming languages are process algebras which are languages that consists of a small set of well-designed operators that describe the interaction and communication between concurrent programs.

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 nondeterministic GSOS format [BIM95]. The nondeterministic GSOS format has been successfully applied to specify many nondeterministic process algebras, most prominently CCS [Mil89] and CSP [BHR84]. In the

same vein, the probabilistic GSOS format allows to specify probabilistic nondeterministic process algebras, such as probabilistic CCS [JLY01; Bar04; DD07], probabilistic CSP [JLY01; Bar04; Den+07; DL12] and probabilistic ACP [And99; And02].

Definition 2.35 (PGSOS rule, [Bar04; LGD12])**.** A *PGSOS rule* r has the form:

$$\frac{\{x_i \xrightarrow{a_{i,k}} \mu_{i,k} \mid i \in I, k \in K_i\} \qquad \{x_i \xrightarrow{b_{i,l}} \not\to \mid i \in I, l \in L_i\}}{f(x_1, \ldots, x_n) \xrightarrow{a} \theta}$$

with $f \in \Sigma$ an operator with rank n, $I = \{1, \ldots, n\}$ 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 $x_1, \ldots, x_n$ are pairwise different;
3. $\mathsf{Var}(\theta) \subseteq \{\mu_{i,k} \mid i \in I, k \in K_i\} \cup \{x_1 \ldots, x_n\}$.

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 2.36 (Notations for rules)**.** Let r be a PGSOS rule. The expressions $x_i \xrightarrow{a_{i,k}} \mu_{i,k}$, $x_i \xrightarrow{b_{i,l}} \not\to$ and $f(x_1, \ldots, x_n) \xrightarrow{a} \theta$ are called, resp., *positive premises*, *negative premises* and *conclusion*. The set of all premises is denoted by $\mathrm{prem}(r)$ and the conclusion by $\mathrm{conc}(r)$. The term $f(x_1, \ldots, x_n)$ is called the *source* (notation $\mathrm{src}(r)$), the variables $x_1, \ldots, x_n$ are called *source variables* (notation $x_i \in \mathrm{src}(r)$), and the distribution term θ is called the *target* (notation $\mathrm{trgt}(r)$). Let $\mathrm{der}(r, x_i) = \{\mu_{i,k} \mid x_i \xrightarrow{a_{i,k}} \mu_{i,k} \in \mathrm{prem}(r)\}$ denote the variables derived from the source variable x_i. We call $\mu \in \mathrm{der}(r, x_i)$ a *derivative* of x_i.

Given a set of rules R we denote by R_f the rules specifying operator f, i.e. all rules of R with source $f(x_1, \ldots, x_n)$, 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 2.37 (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 (Defintion 2.39) is image-finite such that the fixed point characterization of bisimulation metrics coincides with the coinductive characterization (Proposition 2.25).

The operational semantics of terms is given by inductively applying the respective PGSOS rules. Then, a supported model of a specification 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 2.38 (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 $x_i \xrightarrow{a_{i,k}} \mu_{i,k} \in \mathrm{prem}(r)$,
- $\sigma(x_i) \xrightarrow{b_{i,l}} \pi \notin \rightarrow$ for any $\pi \in \Delta(\mathsf{T}(\Sigma))$, for all $x_i \not\xrightarrow{b_{i,l}} \in \mathrm{prem}(r)$, and
- $t \xrightarrow{a} \pi \in \rightarrow$ for $t \xrightarrow{a} \pi = \sigma(\mathrm{conc}(r))$.

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 specification P form the supported model of P.

Definition 2.39 (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 specification P also the *induced model* of P.

Intuitively, a term $f(t_1, \ldots, t_n)$ represents the composition of terms $t_1, \ldots, t_n$ by operator f. A rule r specifies some transition $f(t_1, \ldots, t_n) \xrightarrow{a} \pi$ that represents the evolution of the composed term $f(t_1, \ldots, t_n)$ by action a to the distribution π. We say that a rule with conclusion $f(x_1, \ldots, x_n) \xrightarrow{a} \theta$ *delays* the evolution of the source term x_i if x_i appears in the rule target θ, and that the source term x_i *evolves* to $\mu \in \mathrm{der}(r, x_i)$ if μ appears in the rule target θ. Note that a rule may both delay and evolve a source term. We say that r *replicates* a source variable x_i if multiple instances of either x_i or x_i-derivatives in $\mathrm{der}(r, x_i)$ appear in the target θ of rule r.

Example 2.40. We reconsider the language defined in Example 2.11. Let $P = (\Sigma, A, R)$ be a PTSS with the following rules in R:

$$\frac{}{s \xrightarrow{a} \delta(s)} \qquad \frac{}{t \xrightarrow{a} 0.5\delta(t) + 0.5\delta(0)} \qquad \frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu}{x \mid y \xrightarrow{a} \mu \mid \nu}$$

We refer to the three rules by r_1, r_2, r_3. It is easy to verify that all rules are in PGSOS format. Let $M = (\mathsf{T}(\Sigma), A, \rightarrow)$ be the induced model of P.

The rules r_1 and r_2 are called axioms since they have no premises. In this case, any instance of the conclusion is a supported transition of P. Hence, we get $\{s \xrightarrow{a}$

$\delta(s), t \xrightarrow{a} 0.5\delta(t) + 0.5\delta(0)\} \subseteq \rightarrow$. Now we apply the third rule to derive another supported transition. Let σ be a substitution defined by $\sigma(x) = s$, $\sigma(y) = t$, $\sigma(\mu) = \delta(s)$, $\sigma(\nu) = 0.5\delta(t) + 0.5\delta(0)$. Now we derive $M \models^{\sigma}_{r_3} s \mid t \xrightarrow{a} 0.5\delta(s \mid t) + 0.5\delta(s \mid 0)$. Another transition may be derived by using the substitution $\sigma(x) = t$, $\sigma(y) = t$, $\sigma(\mu) = 0.5\delta(t)+0.5\delta(0)$, $\sigma(\nu) = 0.5\delta(t)+0.5\delta(0)$. Then $M \models^{\sigma}_{r_3} t \mid t \xrightarrow{a} 0.25\delta(t \mid t)+0.25\delta(t \mid 0) + 0.25\delta(0 \mid t) + 0.25\delta(0 \mid 0)$. Finally, let us remark that there are many more states, e.g. $(s \mid s) \mid t$, $(t \mid t) \mid s$, $(t \mid t) \mid t$, with transitions defined by the rules in R.

We will study specifications where not all operators satisfy the same compositionality property. In order to exploit the particular compositionality property of each specified operator, we will stratify the specifications based on the strength of the compositionality property of the specified operators. We formalize this by using the concept of disjoint extension.

Definition 2.41 (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. We may write $P_2 \sqsupseteq P_1$ for $P_1 \sqsubseteq P_2$.

Example 2.42. Consider the PTSS $P = (\Sigma, A, R)$ specified in Example 2.40. The rules in R specify the operator $\mid$ as synchronous parallel composition operator. Assume that we would like to specify that the process $s \mid t$ may evolve both synchronously and asynchronously. Then, we need to specify additionally the following rules:

$$\frac{x \xrightarrow{a} \mu}{x \mid y \xrightarrow{a} \mu \mid \delta(y)} \qquad \frac{y \xrightarrow{a} \mu}{x \mid y \xrightarrow{a} \delta(x) \mid \mu}$$

Let R' be the set with these two rules, $R_2 = R \cup R'$ and $\Sigma_2 = \Sigma$.

Note that the specification $P_2 = (\Sigma_2, A, R_2)$ is not a disjoint extension of P since R_2 introduces new rules R' for the operator $\mid \in \Sigma$. On the operational semantics level, we have $M \models_{P_2} s \mid t \xrightarrow{a} \delta(s) \mid \delta(t)$ but $M \not\models_{P} s \mid t \xrightarrow{a} \delta(s) \mid \delta(t)$. In other words, the rules of P_2 introduce new transitions on terms of P. Those new transition on terms of P may change the compositionality properties of those terms. In other words, if an extension is not disjoint, then the extension may not preserve the compositionality properties of the already specified operators. The concept of disjoint extensions allows us to extend specifications without changing the operational semantics of already specified operators and terms. In other words, disjoint extensions preserve the compositionality properties of already specified operators. We remark that the more general concept of conservative extension [GV92; AFV01a; DGL15] would already suffice to preserve the compositionality properties. However, the technically simpler concept of disjoint extension is sufficient for all practically relevant cases and allows for simpler argumentation and proofs.

A possible approach to model that process s and t evolves both synchronously and asynchronously is to introduce a new operator for asynchronous composition and an operator for the choice between both behavior. Let R'' be a set with the following rules:

$$\frac{x \xrightarrow{a} \mu}{x \mid\mid\mid y \xrightarrow{a} \mu \mid\mid\mid \delta(y)} \qquad \frac{y \xrightarrow{a} \nu}{x \mid\mid\mid y \xrightarrow{a} \delta(x) \mid\mid\mid \nu} \qquad \frac{x \xrightarrow{a} \mu}{x + y \xrightarrow{a} \mu} \qquad \frac{y \xrightarrow{a} \nu}{x + y \xrightarrow{a} \nu}$$

Let $\Sigma'' = (\{|||, +\}, \{||| \mapsto 2, + \mapsto 2\})$, $\Sigma_3 = \Sigma \cup \Sigma''$ and $R_3 = R \cup R''$. It is easy to verify that $P \sqsubseteq (\Sigma_3, A, R_3)$. Moreover, the term $(s \mid t) + (s \mid\mid\mid t)$ expresses now a parallel composition between s and t that may evolve synchronously or asynchronously.

Chapter 3

Compositional metric reasoning

3.1 Introduction

Probabilistic process algebras are languages to describe probabilistic concurrent communicating systems (probabilistic processes for short). In this chapter we will study compositional reasoning over probabilistic processes. A probabilistic process is specified by a process term of some probabilistic process algebra. The operational semantics of a process term is a probabilistic nondeterministic transition system with transitions derived from SOS rules in the probabilistic GSOS format.

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 a common agreement that compositional reasoning requires that the considered behavioral equivalence is a congruence w.r.t. all operators. On the other hand, for behavioral metric semantics there are several proposals of properties that operators should satisfy in order to facilitate compositional reasoning. Most prominent examples are non-expansiveness [Des+04] and non-extensiveness [Bac+13]. We discuss these properties and propose uniform continuity as the most natural property of process operators to facilitate compositional reasoning w.r.t. behavioral metric semantics especially in presence of recursion. Uniform continuity 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 bisimulation distance ϵ (understood as the admissible tolerance from the operational behavior of the composed process $f(p_1, p_2)$) there are non-zero bisimulation distances δ_1 and δ_2 (understood as the admissible tolerances from the operational behavior of the processes p_1 and p_2) such that the distance between the composed processes $f(p_1, p_2)$ and $f(p_1', p_2')$ is at most ϵ whenever the component p_1' (resp. p_2') is in distance of at most δ_1 from p_1 (resp. at most δ_2 from p_2).

The main contributions of this chapter are:

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 3.2.2 and 3.3.2).

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. uniform continuity (Section 3.2).

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 uniform continuity, but not w.r.t. non-expansiveness and non-extensiveness (Section 3.3).

4. We demonstrate the usefulness of compositional reasoning using a network protocol built from uniformly continuous operators. In particular, we show how it is possible to derive performance guarantees for the entire system from performance assumptions about individual components. Conversely, we show how it is also possible to derive performance requirements on individual components from performance requirements on the complete system (Section 3.4).

This chapter has been partially published as [GLT15].

3.2 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 [Des+04; Den+05] which considered only a small set of process combinators and only the 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.

3.2.1 Non-recursive process combinators

We introduce a probabilistic process algebra that comprises many of the probabilistic process combinators from CCS [Bar04; DD07] and CSP [DL12; Den+07]. Let Σ_{PA} be a signature with the following operators: i) constants 0 (stop process) and ε (skip process); ii) a family of n-ary probabilistic prefix operators $a.([p_1]\_ \oplus \ldots \oplus [p_n]\_)$ with $a \in A$, $n \geq 1$, $p_1, \ldots, p_n \in (0,1]$ and $\sum_{i=1}^{n} p_i = 1$; iii) binary operators $\_;\_$ (sequential composition), $\_ + \_$ (alternative composition), $\_ +_p \_$ (probabilistic alternative composition), $\_ \mid \_$ (synchronous parallel composition), $\_ \mid\mid\mid \_$ (asynchronous parallel composition), $\_ \mid\mid\mid_p \_$ (probabilistic parallel composition), and $\_ \|_B \_$ for each for each $B \subseteq A$ (CSP 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 3.1 and Table 3.2.

The probabilistic prefix operator expresses that the process $a.([p_1]t_1 \oplus \ldots \oplus [p_n]t_n)$ can perform action a and evolves to process t_i with probability p_i. We write $a.\bigoplus_{i=1}^{n}[p_i]t_i$ for $a.([p_1]t_1 \oplus \ldots \oplus [p_n]t_n)$ and $a.t$ for $a.([1]t)$ (deterministic prefix operator). The

$$\frac{}{\varepsilon \xrightarrow{\surd} \delta(0)} \qquad \frac{}{a.\bigoplus_{i=1}^{n}[p_i]x_i \xrightarrow{a} \sum_{i=1}^{n} p_i\delta(x_i)}$$

$$\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x;y \xrightarrow{a} \mu;\delta(y)} \qquad \frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{a} \nu}{x;y \xrightarrow{a} \nu} \qquad \frac{x \xrightarrow{a} \mu}{x+y \xrightarrow{a} \mu} \qquad \frac{y \xrightarrow{a} \nu}{x+y \xrightarrow{a} \nu}$$

$$\frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu \quad a \neq \surd}{x \mid y \xrightarrow{a} \mu \mid \nu} \qquad \frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{\surd} \nu}{x \mid y \xrightarrow{a} \delta(0)}$$

$$\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x \mid\mid\mid y \xrightarrow{a} \mu \mid\mid\mid \delta(y)} \qquad \frac{y \xrightarrow{a} \nu \quad a \neq \surd}{x \mid\mid\mid y \xrightarrow{a} \delta(x) \mid\mid\mid \nu} \qquad \frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{\surd} \nu}{x \mid\mid\mid y \xrightarrow{\surd} \delta(0)}$$

$$\frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu \quad a \in B \setminus \{\surd\}}{x \mid\mid_B y \xrightarrow{a} \mu \mid\mid_B \nu} \qquad \frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{\surd} \nu}{x \mid\mid_B y \xrightarrow{\surd} \delta(0)}$$

$$\frac{x \xrightarrow{a} \mu \quad a \notin B \cup \{\surd\}}{x \mid\mid_B y \xrightarrow{a} \mu \mid\mid_B \delta(y)} \qquad \frac{y \xrightarrow{a} \nu \quad a \notin B \cup \{\surd\}}{x \mid\mid_B y \xrightarrow{a} \delta(x) \mid\mid_B \nu}$$

Table 3.1: Standard non-recursive process combinators

sequential composition and the alternative composition are as usual. The synchronous parallel composition $t \mid t'$ describes the simultaneous evolution of processes t and t', while the asynchronous parallel composition $t \mid\mid\mid t'$ describes the interleaving of t and t' where both processes can progress by alternating at any rate the execution of their actions. The CSP-like parallel composition $t \mid\mid_B t'$ 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 $t \mid\mid\mid_p t'$ evolves to a probabilistic choice between the nondeterministic choices of $t \mid\mid\mid t'$.

3.2.2 Distance between non-recursive processes

We develop now tight bounds on the distance between processes combined by the non-recursive process combinators presented in Table 3.1 and Table 3.2. This will allow us to

$$\frac{x \xrightarrow{a} \mu \quad y \not\xrightarrow{a}}{x +_p y \xrightarrow{a} \mu} \qquad \frac{x \not\xrightarrow{a} \quad y \xrightarrow{a} \nu}{x +_p y \xrightarrow{a} \nu} \qquad \frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu}{x +_p y \xrightarrow{a} \mu \oplus_p \nu}$$

$$\frac{x \xrightarrow{a} \mu \quad y \not\xrightarrow{a} \quad a \neq \surd}{x \,|||_p\, y \xrightarrow{a} \mu \,|||_p\, \delta(y)} \qquad \frac{x \not\xrightarrow{a} \quad y \xrightarrow{a} \nu \quad a \neq \surd}{x \,|||_p\, y \xrightarrow{a} \delta(x) \,|||_p\, \nu}$$

$$\frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu \quad a \neq \surd}{x \,|||_p\, y \xrightarrow{a} \mu \,|||_p\, \delta(y) \oplus_p \delta(x) \,|||_p\, \nu} \qquad \frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{\surd} \nu}{x \,|||_p\, y \xrightarrow{\surd} \delta(0)}$$

Table 3.2: Standard non-recursive probabilistic process combinators

derive the compositionality properties of those operators. As we will discuss two different compositionality properties for non-recursive processes, 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 $f(s_1,\ldots,s_n)$ and $f(t_1,\ldots,t_n)$ in terms of the distance between their respective components s_i and t_i. Intuitively, given a probabilistic process $f(s_1,\ldots,s_n)$ we provide a bound on the distance to the respective probabilistic process $f(t_1,\ldots,t_n)$ 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 $s_i, t_i \in T(\Sigma)$ 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. We start with the probabilistic prefix operator (Proposition 3.1.a). There are the transitions $a.\bigoplus_{i=1}^n [p_i]s_i \xrightarrow{a} \sum_{i=1}^n p_i \delta(s_i)$ 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 $\lambda \cdot \mathbf{K}(\mathbf{d})(\sum_{i=1}^n p_i\delta(s_i), \sum_{i=1}^n p_i\delta(t_i)) \leq \lambda \cdot \sum_{i=1}^n p_i \mathbf{d}(s_i,t_i)$. By Proposition 2.32 we obtain the following inequations:

$$\begin{aligned}
&\mathbf{K}(\mathbf{d})\left(\sum_{i=1}^n p_i\delta(s_i), \sum_{i=1}^n p_i\delta(t_i)\right) \\
&\leq \sum_{i=1}^n p_i \,\mathbf{K}(\mathbf{d})(\delta(s_i),\delta(t_i)) && \text{(Proposition 2.32.3)} \\
&\leq \sum_{i=1}^n p_i \mathbf{d}(s_i,t_i) && \text{(Proposition 2.32.2)}
\end{aligned}$$

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. If $s_1+s_2 \xrightarrow{a} \pi$ is derived from $s_1 \xrightarrow{a} \pi$, then, 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. If $s_1+s_2 \xrightarrow{a} \pi$ is derived from $s_2 \xrightarrow{a} \pi$ then the argument is analogous.

We conclude with 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 metric bisimulation transfer condition, 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 2.30.2 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 $t_2 \xrightarrow{a} \pi_2'$ 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

$$
\begin{aligned}
&\lambda \cdot \mathbf{K}(\mathbf{d})(p\pi_1+(1-p)\pi_2, p\pi_1'+(1-p)\pi_2') \\
\leq &\lambda \cdot (p\,\mathbf{K}(\mathbf{d})(\pi_1,\pi_1') + (1-p)\,\mathbf{K}(\mathbf{d})(\pi_2,\pi_2')) && \text{(Proposition 2.32.3)} \\
\leq &\lambda \cdot \max(\mathbf{K}(\mathbf{d})(\pi_1,\pi_1'), \mathbf{K}(\mathbf{d})(\pi_2,\pi_2')) \\
\leq &\max(\mathbf{d}(s_1,t_1),\mathbf{d}(s_2,t_2)).
\end{aligned}
$$

□

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 defines a convex combination of these transitions. If the probabilistic alternative composition would be defined by only the third rule of Table 3.2, then $\mathbf{d}(s_1 +_p s_2, t_1 +_p t_2) \leq p\mathbf{d}(s_1,t_1)+(1-p)\mathbf{d}(s_2,t_2)$.

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

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

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

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

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

with

$$d^s = \begin{cases} 1 & \text{if } \mathbf{d}(s_1,t_1)=1 \\ 1 & \text{if } \mathbf{d}(s_2,t_2)=1 \\ \mathbf{d}(s_1,t_1)+(1-\mathbf{d}(s_1,t_1)/\lambda)\mathbf{d}(s_2,t_2) & \text{otherwise} \end{cases}$$

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

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

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

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 $\mid$ coincides with $\parallel_A$ and $\mid\mid\mid$ coincides with $\parallel_\emptyset$. The proofs for the probabilistic parallel composition operator $\mid\mid\mid_p$ (Proposition 3.2.e) and the sequential composition ; (Proposition 3.2.a) are analogous.

We consider $B \setminus \{\surd\} \neq \emptyset$ (the case $B \setminus \{\surd\} = \emptyset$ is similar). First we need to introduce the notion of congruence closure for $\mathbf{d}$ 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 $\|_B$ w.r.t. the bound provided in Proposition 3.2.d as a function $d : \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \to [0,1]$ defined by

$$d(t,t') = \begin{cases} \min(\lambda[1-(1-d(t_1,t_1')/\lambda)(1-d(t_2,t_2')/\lambda)], \mathbf{d}(t,t')) & \text{if } \left[\begin{array}{l} t = t_1 \|_B t_2 \wedge \\ t' = t_1' \|_B t_2' \wedge \\ \mathbf{d}(t_1,t_1') < 1 \wedge \\ \mathbf{d}(t_2,t_2') < 1 \end{array}\right. \\ \mathbf{d}(t,t') & \text{otherwise} \end{cases}$$

We note that d satisfies by construction $d(s_1 \|_B s_2, t_1 \|_B t_2) \leq d^s$ 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

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

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

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

Consider the base case $k = 0$. 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 $\pi' \in \Delta(\mathsf{T}(\Sigma))$ such that $\lambda \cdot \mathbf{K}(\mathbf{d})(\pi,\pi') \leq \mathbf{d}(t,t')$. By Proposition 2.32 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 $t = t_1 \|_B t_2$, or t' is not of the form $t' = t_1' \|_B t_2'$, we have $d(t,t') = \mathbf{d}(t,t')$ and Equation 3.1 follows precisely as in the base case $k = 0$. If both $t = t_1 \|_B t_2$ and $t' = t_1' \|_B t_2'$, then we distinguish two cases, namely $d(t,t') = \mathbf{d}(t,t')$ (either $\mathbf{d}(t_1,t_1') = 1$ or $\mathbf{d}(t_2,t_2') = 1$) 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 $\mathbf{d}(t_2,t_2') < 1$). 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 $\pi = \pi_1 \|_B \pi_2$;
2. $t_1 \xrightarrow{a} \pi_1$, $t_2 \not\xrightarrow{a}$, $a \notin B \cup \{\surd\}$ and $\pi = \pi_1 \|_B \delta(t_2)$;
3. $t_2 \xrightarrow{a} \pi_2$, $t_1 \not\xrightarrow{a}$, $a \notin B \cup \{\surd\}$ and $\pi = \delta(t_1) \|_B \pi_2$;

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 2.33 (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 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 2.30.2 we have that $t_2' \overset{a}{\not\rightarrow}$, therefore we can derive the transition $t_1' \parallel_B t_2' \xrightarrow{a} \pi_1' \parallel_B \delta(t_2')$. Then

$$\begin{aligned}
&\lambda \cdot \mathbf{K}(d)(\pi_1 \parallel_B \delta(t_2), \pi_1' \parallel_B \delta(t_2')) \\
\leq &\lambda^2[1-(1-\mathbf{K}(d)(\pi_1, \pi_1')/\lambda)(1-\mathbf{K}(d)(\delta(t_2), \delta(t_2'))/\lambda)] \\
\leq &\lambda^2[1-(1-d(t_1, t_1')/\lambda^2)(1-d(t_2, t_2')/\lambda)] \\
\leq &\lambda^2[1-(1-d(t_1, t_1')/\lambda^2)(1-d(t_2, t_2')/\lambda^2)] \\
\leq &\lambda[1-(1-d(t_1, t_1')/\lambda)(1-d(t_2, t_2')/\lambda)] \\
= &d(t_1 \parallel_B t_2, t_1' \parallel_B t_2')
\end{aligned}$$

with step 1 again from Theorem 2.33 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 2.32.2.

The third step 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 $t_1' \xrightarrow{\surd} \pi_1'$ and $t_2' \xrightarrow{\surd} \pi_2'$. Hence, there is also the transition $t_1' \parallel_B t_2' \xrightarrow{\surd} \delta(0)$. 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 is satisfied for d also in this case.

□

The expression d^s 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 distances $\mathbf{d}(s_1, t_1)$ and $\mathbf{d}(s_2, t_2)$ contribute symmetrically to d^s since $\mathbf{d}(s_1, t_1) + (1-\mathbf{d}(s_1, t_1)/\lambda)\mathbf{d}(s_2, t_2) = \mathbf{d}(s_2, t_2) + (1-\mathbf{d}(s_2, t_2)/\lambda)\mathbf{d}(s_1, t_1)$.

The expressions $d^a_{1,2}, d^a_{2,1}, d^a$ cover different scenarios of the asynchronous evolution of those processes. The expression $d^a_{1,2}$ (resp. $d^a_{2,1}$) 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 transition is performed by the processes s_1 and t_1 (resp. the first transition is performed by processes s_2 and t_2). Finally, d^a captures the distance between asynchronously evolving processes independent of which of those processes moves first.

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. Remark 2.31). We analyze the bound for the process combinator 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) the distance $d^a_{1,2}$, which captures the case that first the processes s_1 and t_1 evolve followed by s_2 and t_2, and (ii) the 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. The distance $d^a_{1,2}$ weights the distance between s_2 and t_2 by $\lambda(1 - \mathbf{d}(s_1, t_1)/\lambda)$. The discount λ expresses that the distance between processes s_2 and t_2 is observable just after s_1 and t_1 have performed at least one step. 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 $t_1 \mid t_2$ is the expression d^s, which is $\mathbf{d}(s_1, t_1) + (1 - \mathbf{d}(s_1, t_1)/\lambda)\mathbf{d}(s_2, t_2) = \mathbf{d}(s_2, t_2) + (1 - \mathbf{d}(s_2, t_2)/\lambda)\mathbf{d}(s_1, t_1) = \lambda(1 - (1 - \mathbf{d}(s_1, t_1)/\lambda)(1 - \mathbf{d}(s_2, t_2)/\lambda))$ 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, 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 expressed by $1 - (1 - \mathbf{d}(s_1, t_1)/\lambda)(1 - \mathbf{d}(s_2, t_2)/\lambda)$ multiplied by the discount factor λ.

The distance bound between asynchronous parallel composed processes $s_1 \mid\mid\mid s_2$ and $t_1 \mid\mid\mid t_2$ is the expression d^a. Hence the distance bound is the maximum of $d^a_{1,2}$, namely the distance observable when first processes s_1 and t_1 evolve and then s_2 and t_2, and $d^a_{2,1}$, namely the distance observable when first processes s_2 and t_2 evolve and then s_1 and t_1. Both $d^a_{1,2}$ and $d^a_{2,1}$ differ from the distance d^s of the synchronously evolving processes only by the discount factor λ that is applied to the distance of the delayed processes.

The distance between processes composed by the probabilistic parallel composition operator $s_1 \mid\mid\mid_p s_2$ and $t_1 \mid\mid\mid_p t_2$ is bounded by the same 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.

Processes that are composed by the CSP parallel composition operator $\_ \|_B \_$ 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 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 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 $s_i, t_i \in T(\Sigma_{PA})$ with $\mathbf{d}(s_i, t_i) = \epsilon_i$ such that the inequalities in Propositions 3.1 and 3.2 become equalities.*

Proof. Consider Proposition 3.1. The maximal distance is realized if the composed processes perform different initial actions. Let $A = \{a_i \mid 1 \leq i \leq n\} \cup \{\surd\}$. We define

- $s_i = t_i = a_i.\varepsilon$, 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 $\epsilon_i = \lambda$;
- and $s_i = 0$ and $t_i = a_i.\varepsilon$, if $\epsilon_i = 1$.

These processes yield for all process combinators of Proposition 3.1 exactly the stated upper bound.

Consider Proposition 3.2. The maximal distance is realized if the composed processes may perform the same actions and can synchronize. Let $A = \{a, \surd\}$. We define

- $s_i = t_i = a.\varepsilon$, 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 $\epsilon_i = \lambda$;
- $s_i = 0$ and $t_i = a.\varepsilon$ if $\epsilon_i = 1$.

These processes yield for all process combinators of Proposition 3.2 exactly the stated upper bound. □

3.2.3 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 [Bac+13] and non-expansiveness [Des+02a; Des+04; Den+05; Cha+14]), 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 its parts. Later in Section 3.3.3 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

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

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 (cf. Proposition 3.2 and 3.3).

Proposition 3.6. *None of the process combinators*

- *sequential composition* $\_ ; \_$
- *synchronous parallel composition* $\_ \mid \_$
- *asynchronous parallel composition* $\_ \mid\mid\mid \_$
- *CSP-like parallel composition* $\_ \parallel_B \_$
- *probabilistic parallel composition* $\_ \mid\mid\mid_p \_$

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

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

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.

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

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

Theorem 3.8 generalizes a similar result of [Des+04] which considered only PTSs without nondeterministic 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) are novel.

3.3 Recursive processes

Recursion is necessary to express infinite (non-terminating) behavior in terms of finite process expressions. Moreover, recursion allows 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 used for non-recursive process combinators (Section 3.2.3) fall short for recursive process combinators. We will propose the more general property of uniform continuity (Section 3.3.3) that captures the inherent nature of compositional reasoning over probabilistic processes. In fact, it allows 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.

3.3.1 Recursive process combinator

We define P_{PA° as disjoint extension of P_{PA} with the 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 3.3.

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 variants of Kleene-star iteration [Bar04, Section 5.2.4(vi)] and bang replication [MS13, Fig. 1] substitute the nondeterministic choice of the non-probabilistic variants by a respective probabilistic choice.

$$\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x^{n+1} \xrightarrow{a} \mu; \delta(x^n)} \qquad \frac{x \xrightarrow{\surd} \mu}{x^{n+1} \xrightarrow{\surd} \mu} \qquad \frac{}{x^0 \xrightarrow{\surd} \delta(0)} \qquad \frac{x \xrightarrow{\surd} \mu \quad x \xrightarrow{a} \nu \quad a \neq \surd \quad n > m}{x^n \xrightarrow{a} \nu; \delta(x^m)}$$

$$\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x^\omega \xrightarrow{a} \mu; \delta(x^\omega)} \qquad \frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x^* y \xrightarrow{a} \mu; \delta(x^* y)} \qquad \frac{y \xrightarrow{a} \nu}{x^* y \xrightarrow{a} \nu}$$

$$\frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu \quad a \neq \surd}{x^{*_p} y \xrightarrow{a} \nu \oplus_p \mu; \delta(x^{*_p} y)} \qquad \frac{x \xrightarrow{a} \mu \quad y \not\xrightarrow{a} \quad a \neq \surd}{x^{*_p} y \xrightarrow{a} \mu; \delta(x^{*_p} y)} \qquad \frac{x \not\xrightarrow{a} \quad y \xrightarrow{a} \nu \quad a \neq \surd}{x^{*_p} y \xrightarrow{a} \nu} \qquad \frac{y \xrightarrow{\surd} \nu}{x^{*_p} y \xrightarrow{\surd} \nu}$$

$$\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{!^{n+1} x \xrightarrow{a} \mu \,|||\, \delta(!^n x)} \qquad \frac{x \xrightarrow{\surd} \mu}{!^{n+1} x \xrightarrow{\surd} \mu} \qquad \frac{}{!^0 x \xrightarrow{\surd} \delta(0)}$$

$$\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{!x \xrightarrow{a} \mu \,|||\, \delta(!x)} \qquad \frac{x \xrightarrow{a} \mu \quad a \neq \surd}{!_p x \xrightarrow{a} \mu \oplus_p (\mu \,|||\, \delta(!_p x))}$$

Table 3.3: Standard recursive process combinators

3.3.2 Distance between recursive processes

We develop now tight bounds for recursive process combinators.

Proposition 3.9. *Let $P = (\Sigma, A, R)$ be any PTSS with $P_{PA^\diamond} \sqsubseteq P$. 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) $\mathbf{d}(!s, !t) \leq d^\omega$

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

with

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

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

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

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 can 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 3.1–3.3, 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 $\mathbf{d}(!^n s,!^n t) \leq \mathbf{d}(s,t)\sum_{k=0}^{n-1}(\lambda-\mathbf{d}(s,t))^k$. 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 3.1–3.3, we infer that $!^{n+1}s$ is bisimilar to $s \mid\mid\mid !^n s$ and that $!^{n+1}t$ is bisimilar to $t \mid\mid\mid !^n t$. Hence $\mathbf{d}(!^{n+1}s,!^{n+1}t) = \mathbf{d}(s \mid\mid\mid !^n s, t \mid\mid\mid !^n t)$. By Proposition 3.2.c we get $\mathbf{d}(s \mid\mid\mid !^n s, t \mid\mid\mid !^n t) \leq \mathbf{d}(s,t)+(\lambda-\mathbf{d}(s,t))\mathbf{d}(!^n s,!^n t) \leq$ (inductive hypothesis) $\mathbf{d}(s,t)+(\lambda-\mathbf{d}(s,t))\mathbf{d}(s,t)\sum_{k=0}^{n-1}(\lambda-\mathbf{d}(s,t))^k = \mathbf{d}(s,t)\sum_{k=0}^{n}(\lambda-\mathbf{d}(s,t))^k$. Summarizing, we have $\mathbf{d}(!^{n+1}s,!^{n+1}t) \leq \mathbf{d}(s,t)\sum_{k=0}^{n}(\lambda-\mathbf{d}(s,t))^k$. 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 3.1–3.3, 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 we get $\mathbf{d}(s;s^\omega,t;t^\omega) \leq \mathbf{d}(s,t)+(\lambda-\mathbf{d}(s,t))\mathbf{d}(s^\omega,t^\omega)$. From $\mathbf{d}(s^\omega,t^\omega) \leq \mathbf{d}(s,t)+(\lambda-\mathbf{d}(s,t))\mathbf{d}(s^\omega,t^\omega)$ 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 3.1–3.3, we infer that $!s$ is bisimilar to $s \mid\mid\mid !s$ and that $!t$ is bisimilar to $t \mid\mid\mid !t$. Hence $\mathbf{d}(!s,!t) = \mathbf{d}(s \mid\mid\mid !s, t \mid\mid\mid !t)$. By Proposition 3.2 we get $\mathbf{d}(s \mid\mid\mid !s, t \mid\mid\mid !t) \leq \mathbf{d}(s,t)+(\lambda-\mathbf{d}(s,t))\mathbf{d}(!s,!t)$. By $\mathbf{d}(!s,!t) \leq \mathbf{d}(s,t)+(\lambda-\mathbf{d}(s,t))\mathbf{d}(!s,!t)$ we get $\mathbf{d}(!s,!t) \leq \mathbf{d}(s,t)\frac{1}{1-(\lambda-\mathbf{d}(s,t))} = d^\omega$.

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 $t_1{}^*t_2$ is bisimilar to $(t_1;(t_1{}^*t_2))+t_2$. Then $\mathbf{d}(s_1{}^*s_2,t_1{}^*t_2) = \mathbf{d}((s_1;(s_1{}^*s_2))+s_2,(t_1;(t_1{}^*t_2))+t_2) = \max\{\mathbf{d}((s_1;(s_1{}^*s_2)),(t_1;(t_1{}^*t_2))),\mathbf{d}(s_2,t_2)\}$ by Proposition 3.2. Now $\mathbf{d}((s_1;(s_1{}^*s_2)),(t_1;(t_1{}^*t_2))) = \mathbf{d}(s_1,t_1)+(\lambda-\mathbf{d}(s_1,t_1))\mathbf{d}(s_1{}^*s_2,t_1{}^*t_2)) = \mathbf{d}(s_1,t_1)+(\lambda-\mathbf{d}(s_1,t_1))\max\{\mathbf{d}((s_1;(s_1{}^*s_2)),(t_1;(t_1{}^*t_2))),\mathbf{d}(s_2,t_2)\}$ by Proposition 3.2. Since $s_1{}^*s_2$ is bisimilar to $(s_1;(s_1{}^*s_2))+s_2$ and $t_1{}^*t_2$ is bisimilar to $(t_1;(t_1{}^*t_2))+t_2$ we have $\max\{\mathbf{d}((s_1;(s_1{}^*s_2)),(t_1;(t_1{}^*t_2))),\mathbf{d}(s_2,t_2)\} = \mathbf{d}((s_1;(s_1{}^*s_2)),(t_1;(t_1{}^*t_2)))$. Then we get $\mathbf{d}(s_1{}^*s_2,t_1{}^*t_2) = \mathbf{d}(s_1,t_1)+(\lambda-\mathbf{d}(s_1,t_1))(\mathbf{d}(s_1,t_1)+(\lambda-\mathbf{d}(s_1,t_1))(\mathbf{d}(s_1{}^*s_2,t_1{}^*t_2))))$. Now we get $\mathbf{d}(s_1{}^*s_2,t_1{}^*t_2) = \mathbf{d}(s_1,t_1)\sum_{k=0}^{\infty}(\lambda-\mathbf{d}(s_1,t_1)) = \mathbf{d}(s_1,t_1)\frac{1}{1-(\lambda-\mathbf{d}(s_1,t_1))}$. Summarizing, it follows that $\mathbf{d}(s_1{}^*s_2,t_1{}^*t_2) = \max\{\mathbf{d}((s_1;(s_1{}^*s_2)),(t_1;(t_1{}^*t_2))),\mathbf{d}(s_2,t_2)\} = \max\{\mathbf{d}(s_1,t_1)\frac{1}{1-(\lambda-\mathbf{d}(s_1,t_1))},\mathbf{d}(s_2,t_2)\} = \max\{\mathbf{d}(s_1^\omega,t_1^\omega),\mathbf{d}(s_2,t_2)\}$.

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 $!_p s$ behaves as $!^{n+1}s$ with probability $p(1-p)^n$. Hence, by Proposition 3.9.b we get $\mathbf{d}(!_p s, !_p t) \leq \sum_{n=0}^{\infty} p(1-p)^n \mathbf{d}(!^{n+1}s, !^{n+1}t) \leq \sum_{n=0}^{\infty} p(1-p)^n d^{n+1} = \mathbf{d}(s,t)/(1-(1-p)(\lambda-\mathbf{d}(s,t)))$. □

First we explain the distance bounds for the nondeterministic recursive process combinators. To understand the distance bound between processes that iterate finitely many times (Proposition 3.9.a), observe that s^n and $s;\ldots;s$, with $s;\ldots;s$ 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 $\mathbf{d}(s^n, t^n) = \mathbf{d}(s;\ldots;s, t;\ldots;t) \leq \mathbf{d}(s,t)\sum_{k=0}^{n-1}(\lambda - \mathbf{d}(s,t)) = d^n$. The same reasoning applies to the finite replication operator (Proposition 3.9.b) by observing that $!^n s$ and $s \,|||\, \ldots \,|||\, s$, with $s \,|||\, \ldots \,|||\, s$ denoting n occurrences of s that evolve asynchronously, denote the same PTSs (up to renaming of states) and that the bounds in Proposition 3.2.a and 3.2.c coincide if $s_1 = s_2 = s$ and $t_1 = t_2 = t$. The distance between processes that may iterate infinitely many times (Proposition 3.9.c), and the distance between processes that may spawn infinitely many copies that evolve asynchronously (Proposition 3.9.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 3.9.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 3.9.f), since the second and the third 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 3.9.b we get $\mathbf{d}(!_p s, !_p t) \leq \sum_{n=0}^{\infty} p(1-p)^n \mathbf{d}(!^{n+1}s, !^{n+1}t) \leq \sum_{n=0}^{\infty} p(1-p)^n d^{n+1} = \mathbf{d}(s,t)/(1-(1-p)(\lambda-\mathbf{d}(s,t)))$.

The distance bounds on the distance between processes composed by recursive process combinators (Proposition 3.9) are tight.

Proposition 3.10. *Let $\epsilon_i \in [0,1]$. There are $s_i, t_i \in T(\Sigma_{PA})$ with $\mathbf{d}(s_i,t_i) = \epsilon_i$ such that the inequalities in Proposition 3.9 become equalities.*

Proof. The witness processes of Proposition 3.3 that were used to show Proposition 3.2 suffice. □

3.3.3 Compositional reasoning over recursive processes

From Propositions 3.9 and 3.10 it follows that none of the recursive process combinators discussed in this section satisfies the compositionality property of non-expansiveness.

Proposition 3.11. *All recursive process combinators of Σ_{PA^ω} (unbounded recursion and bounded recursion with $n \geq 2$) are not non-expansive w.r.t.* $\mathbf{d}$ *for any $\lambda \in (0, 1]$.*

Proof. Follows directly from Propositions 3.9 and 3.10 and the observation that $d^\omega \geq d^n > \mathbf{d}(s, t)$ whenever $0 < \mathbf{d}(s, t) < 1$. □

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 3.12 (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 $\delta_1, \ldots, \delta_n > 0$ such that

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

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

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 $f(s_1, \ldots, s_n)$ 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 $\mathbf{d}(f(s_1, \ldots, s_n), f(t_1, \ldots, t_n)) < \epsilon$. 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.

The distance bounds of Section 3.3.2 allow us to derive that finitely recursing process combinators are uniformly continuous w.r.t. both non-discounted and discounted bisimilarity metric (Theorem 3.13). On the contrary, unbounded recursing process combinators are uniformly continuous only w.r.t. discounted bisimilarity metric (Theorem 3.14 and Proposition 3.15).

Theorem 3.13. *The process combinators*

- *finite iteration* $\_^n$
- *finite replication* $!^n\_$

- *probabilistic replication (bang)* $!_p\_$

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

Proof. For each operator f, we prove the stronger property that there is a constant $K \in \mathbb{R}_{\geq 0}$ (depending on f) such that $\mathbf{d}(f(s_1,\ldots,s_n),f(t_1,\ldots,t_n)) \leq K \sum_{i=1}^{n} \mathbf{d}(s_i,t_i)$. (This is the well-known Lipschitz continuity, which implies uniform continuity by $\delta_i = \epsilon/(n \cdot K)$.) For finite iteration and finite replication operators, this follows directly from Propositions 3.9.a and 3.9.b, respectively, and the observation that $\frac{1-(\lambda-\mathbf{d}(s,t))^n}{1-(\lambda-\mathbf{d}(s,t))} \leq n = K$. For the probabilistic bang operator it follows from Proposition 3.9.g and the observation $\frac{1}{1-(1-p)(\lambda-\mathbf{d}(s,t))} \leq \frac{1}{1-(1-p)\lambda} = K$. □

Note that the probabilistic bang operator is uniformly 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 3.14. *The process combinators*

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

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

Proof. Also in this case, for each of the operators, we prove the stronger property that there is a $K \in \mathbb{R}_{\geq 0}$ such that $\mathbf{d}(f(s_1,\ldots,s_n),f(t_1,\ldots,t_n)) \leq K \sum_{i=1}^{n} \mathbf{d}(s_i,t_i)$. This follows directly from Proposition 3.9.c–3.9.f and the observation that $\frac{1}{1-(\lambda-\mathbf{d}(s,t))} \leq \frac{1}{1-\lambda} = K$. □

Proposition 3.15. *None of the process combinators*

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

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

Proof. Follows directly from Propositions 3.9 and 3.10. 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 $\mathbf{d}_1(s,t) < \delta$ we have $\mathbf{d}_1(s^\omega,t^\omega) < \epsilon$. We will show this by contradiction. Assume there is some $\delta > 0$. Consider $s = a.([1-\delta/2]\varepsilon \oplus [\delta/2]0)$ and $t = a.\varepsilon$. We have $\mathbf{d}_1(s,t) = \delta/2 < \delta$ and $\mathbf{d}_1(s^\omega,t^\omega) = 1 > \epsilon$. Contradiction. Similar reasoning applies also to the other process combinators. □

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

$$RC(N,T,p,q) = \Bigg[\sum_{0 \le n \le N, n=2k} i(n).\Big(CH(0,T,p,q)\,;\, CH(1,T,p,q) \Big)^{\frac{n}{2}} + \sum_{0 \le n \le N, n=2k+1} i(n).\Big(\Big(CH(0,T,p,q)\,;\, CH(1,T,p,q) \Big)^{\frac{n-1}{2}}\,;\, CH(0,T,p,q) \Big) \Bigg]; res(\mathrm{OK}).\varepsilon$$

$$CH(b,t,p,q) = \sum_{d \in D} i(d).CH'(d,b,t,p,q)$$

$$CH'(d,b,t,p,q) = \begin{cases} (\bot.CH'(d,b,t-1,p,q)) \oplus_p (c(d,b).CH_2(d,b,t,p,q)) & \text{if } t > 0 \\ res(NOK) & \text{if } t = 0 \end{cases}$$

$$CH_2(d,b,t,p,q) = \begin{cases} (lost.CH'(d,b,t-1,p,q)) \oplus_q (ack.\varepsilon) & \text{if } t > 0 \\ res(NOK) & \text{if } t = 0 \end{cases}$$

$$TV = \Bigg[\Big(\Big(\sum_{d \in D} c(d,1).(ack.\varepsilon + lost.\varepsilon) \Big)^* \Big(\sum_{d \in D} c(d,0).o(d).(ack.\varepsilon + lost.\varepsilon) \Big) \Big); \Big(\Big(\sum_{d \in D} c(d,0).(ack.\varepsilon + lost.\varepsilon) \Big)^* \Big(\sum_{d \in D} c(d,1).o(d).(ack.\varepsilon + lost.\varepsilon) \Big) \Big) \Bigg]^\omega$$

Figure 3.1: Specification of the Bounded Retransmission Protocol

3.4 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 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, $d_0, \ldots, d_{n-1}$, 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, which may also get lost, with probability q. If the RC does not receive the acknowledgment for datum d_i within a given time, it assumes that d_i got lost and retries to transmit it. However, the maximal number of attempts for d_i is T. Since also the acknowledgment 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 d_{2k+1} get control bit 1 attached.

The BRP is specified in Fig. 3.1. 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 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. $BRP(N,T,p,q)$ 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. $RC(N,T,p,q)$ 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 by means of action $i(d)$ and tries to send it to the TV. If all 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 emits $res(NOK)$. If TV receives a pair (d,b) by action $c(d,b)$ then, if d is original, namely b is the expected control bit, then d is sent to other TV components by $o(d)$, 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 this implementation and the specification with perfect channels $BRP(N,T,0,0)$.

Proposition 3.16. *Let $N,T \in \mathbb{N}$ and $p,q \in [0,1]$.*

1. *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 (i.e. $BRP(N,T,p,q)$ 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 $(1-\epsilon)\frac{1-(1-(1-\epsilon)^{1/N})^k}{(1-\epsilon)^{1/N}}$.*
   (d) *The likelihood that at least $n \leq N$ of the N data items are sent and acknowledged is $(1-\epsilon)\frac{1-(1-(1-\epsilon)^{1/n})^{nT}}{(1-\epsilon)^{1/n}}$.*
   (e) *The likelihood that N items are sent and acknowledged is $(1-\epsilon)\frac{1-(1-(1-\epsilon)^{1/N})^{N\cdot T}}{(1-\epsilon)^{1/N}}$.*
2. *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 $k \leq T$ retries is $(1-\delta)\cdot\delta^k$.

(c) The likelihood that one datum is sent and acknowledged with at most $k \leq T$ retries is $1-\delta^k$.

Proof. The results in items 1a–1c can be understood by observing that $\epsilon = 1-((1-p)(1-q))^N$ is the likelihood that at least one retry is needed to send the stream of N data, $(1-\epsilon)(1-(1-\epsilon)^{1/N})^k$ is the probability to have k failures in sending or acknowledging a datum together with N successes, and $(1-\epsilon)\frac{1-(1-(1-\epsilon)^{1/N})^k}{(1-\epsilon)^{1/N}} = \sum_{i=0}^{k}(1-\epsilon)(1-(1-\epsilon)^{1/N})^i$. Then, item 1d is item 1c with N instantiated with n and k instantiated with $n\cdot T$, and item 1e is item 1c with k instantiated with $n\cdot T$. The results in item (2) 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, $(1-\delta)\cdot\delta^k = (1-p)(1-q)\cdot(1-(1-p)(1-q))^k$ is the likelihood to have k failures followed by a successful transmission, and $1-\delta^k = \sum_{i=0}^{k}(1-\delta)\cdot\delta^i$. It follows that a channel $CH(b,T,p,q)$ eventually (with possibly up to T retries) succeeds to sent and acknowledge one datum by probability $1-\mathbf{d}_1(CH(b,T,0,0),CH(b,T,p,q))^T$. □

Now we show that by applying the compositionality results given in the previous sections (Propositions 3.1, 3.2, 3.9) we can relate the bisimulation distance between the specification $BRP(N,T,0,0)$ and some implementation $BRP(N,T,p,q)$ of the entire protocol with the distances between the specification and some implementation of its respective components. On the one hand, this allows to derive from specified performance properties of the entire protocol individual performance requirements of its components (compositional verification). On the other hand, it allows to infer from performance properties of the protocol components suitable performance guarantees on the entire protocol (compositional specification).

Proposition 3.17. *Let $N,T \in \mathbb{N}$ and $p,q \in [0,1]$. For all $d \in D$ and $b \in \{0,1\}$ it holds*

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

(b) $\mathbf{d}(CH(b,T,0,0),CH(b,T,p,q)) = 1-(1-p)(1-q)$.

Proof. Case (a) follows from Propositions 3.1, 3.2 and 3.9 and case (b). Case (b) follows directly from Propositions 3.1 and 3.2. Moreover, by combining (a) and (b), we can infer that $\mathbf{d}(BRP(N,T,p,q),BRP(N,T,0,0)) \leq 1-((1-p)(1-q))^N$. □

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 3.16 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, 3.13 and Proposition 3.17 we can derive that $\mathbf{d}(BRP(N,T,p,q),BRP(N,T,0,0)) \leq N/2\cdot(\mathbf{d}(CH(0,T,p,q),CH(0,T,0,0))+\mathbf{d}(CH(1,T,p,q),CH(1,T,0,0))) \leq N(1-(1-p)(1-q))$. Then we infer the following result.

Proposition 3.18. *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:

$$
\begin{aligned}
&\mathbf{d}(BRP(N,T,p,q), BRP(N,T,0,0)) \\
\leq &\mathbf{d}(RC(N,T,p,q), RC(N,T,0,0)) + \mathbf{d}(TV,TV) && \text{(Theorem 3.8)} \\
= &\mathbf{d}(RC(N,T,p,q), RC(N,T,0,0)) \\
\leq &\mathbf{d}((CH(0,T,p,q);CH(1,T,p,q))^{N/2}, (CH(0,T,0,0);CH(1,T,0,0))^{N/2}) && \text{(Theorem 3.5)} \\
\leq &N/2 \cdot \mathbf{d}(CH(0,T,p,q);CH(1,T,p,q), CH(0,T,0,0);CH(1,T,0,0)) && \text{(Theorem 3.13)} \\
\leq &N/2 \cdot (\mathbf{d}(CH(0,T,p,q), CH(0,T,0,0)) + \mathbf{d}(CH(1,T,p,q), CH(1,T,0,0))) && \text{(Theorem 3.8)} \\
= &N(1-(1-p)(1-q)).
\end{aligned}
$$

The case that N is odd is analogous. From $\mathbf{d}(BRP(N,T,p,q), BRP(N,T,0,0)) \leq N(1-(1-p)(1-q))$ the thesis follows. □

Combining Propositions 3.16 – 3.18 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 3.19. 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 3.16.1 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 3.17.a 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 3.17.b 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 3.16.2 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%'.

3.5 Closing remarks

In this chapter we argued that uniform continuity (Definition 3.12, generalizing 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, 3.13, 3.14). In addition, we provided for all standard process algebra operators tight bounds on the distance between the composed processes (Propositions 3.1, 3.2, 3.9). We exemplified how these results can be used to reason compositionally over protocols. In fact, they allow 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 [Des+02a].

The metric reasoning approach exemplified in Section 3.4 is a sound method to reason compositionally over systems. However, the distance between composed systems might not be tight. Let $p[x]$ be an open term describing a composed system with x the placeholder for a subsystem. Given subsystems s and t, the distance $\mathbf{d}(p[s], p[t])$ might be below the composition of the compositionality properties of the operators in p if some of the differences in the behaviors between s and t do not induce different behaviors between $p[s]$ and $p[t]$. To exemplify this effect, consider the context $p[x] = x \mid b.0$ and subsystems $s = a.0$ and $t = a.([1-\epsilon/\lambda]\varepsilon \oplus [\epsilon_i/\lambda]0)$. Clearly $\mathbf{d}(s,t) = \epsilon$. Then the compositional analysis gives $\mathbf{d}(p[s], p[t]) \leq \epsilon$. However, $\mathbf{d}(p[s], p[t]) = 0$ because the behavioral distance between s and t (observable only after executing action a) cannot be observed in the context $p[x]$ (which can only perform an action if the instances of x perform action b). Thus, $\mathbf{d}(p[s], p[t]) = 0$ since s and t 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 p, the p-bisimulation distance (bisimulation distance w.r.t. context p) between s and t would measure only that degree of the bisimulation distance between s and t that would induce different behavior between p instantiated by s and p instantiated by t. Using the notation $\mathbf{d}_p$ for the p-bisimulation distance this would give the behavioral distance $\mathbf{d}_p(s,t) = 0$ (since p derives only behavior from an initial b-move and s and t agree on their inability to perform b-moves), while $\mathbf{d}_p(s = b.0, b.([1-\epsilon/\lambda]\varepsilon \oplus [\epsilon_i/\lambda]0) = \epsilon$. 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.

Chapter 4

Specification of compositional operators

4.1 Introduction

We will generalize now the compositionality results of the former chapter (which were developed for some concrete probabilistic process algebra) to arbitrary probabilistic programming languages and probabilistic process algebras. The compositionality property of some language operator is given as a modulus of continuity that relates the distance between composed processes to the distance between their parts. We explore and generalize the earlier discussed compositionality properties of non-extensiveness (Definition 3.4), non-expansiveness (Definition 3.7), Lipschitz continuity, and uniform continuity (Definition 3.12). Figure 4.1 shows the whole spectrum of compositionality properties (ordered from the strongest to the most liberal).

We develop for each compositionality property an expressive SOS specification format guaranteeing that the specified operators satisfy the compositionality property. The formats are developed by systematically analyzing which rule and specification patterns define and which do not define operators satisfying the compositionality properties. As a result we obtain (a spectrum of) SOS rule and specification formats that allow us to simultaneously specify operators with different compositionality properties (e.g. one operator is 1-Lipschitz continuous and another operator is 2-Lipschitz continuous) in one SOS specification. To the best of our knowledge, our rule and specification formats are the first that allow to specify simultaneously operators of different compositionality properties. Moreover, each rule and specification format exploits additionally the (possibly different) compositionality guarantees of all operators used in the specification rules. This admits an expressive class of specifications.

A fundamental insight of our study is that the modulus of continuity of the compositionality property defines the maximal process replication behavior that is allowed for the operators of the language. More precisely, while on the one hand operators that are non-extensive (as the most demanding compositionality property) may allow that at most one of the composed processes evolves, on the other hand operators that are uniformly

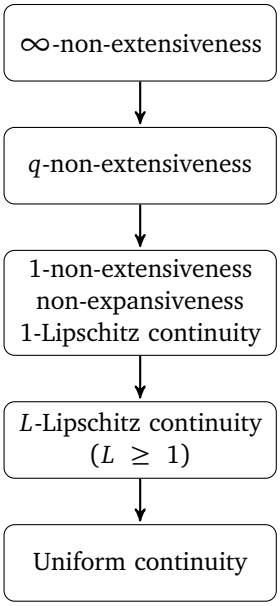

Figure 4.1: Lattice of compositionality properties

continuous (as the most relaxed compositionality property) may allow that finitely many copies of each of the composed processes evolve. In addition, our rule and specification formats provide also novel insights in the interplay between the replication of processes, probabilistic choices between processes, and the (step) discount of the bisimilarity metric. More general, we relate algebraic properties of the bisimilarity metric with structural properties of the specification rules.

Another crucial insight is that an operator is uniformly continuous if it is Lipschitz continuous for each finite projection. The Lipschitz factor of some operator w.r.t. the k-th projection, i.e. w.r.t. the up-to-k bisimilarity metric, is determined by the replication of processes in the first k steps, the probabilistic choices in those steps, and the (step) discount of the bisimulation metric. The SOS specification format derives then from the definition of Lipschitz factors of the finite projections the guarantee that the specified operator is uniformly continuous.

We develop also a coinductive characterization of the spectrum of rule formats. This allows us then to derive from any modulus of continuity the respective syntactic requirements on the specifications ensuring that the specified operators satisfy this modulus of continuity. Furthermore, the coinductive approach allows also to show by direct argumentation that syntactic compositionality (composing syntactic rule formats) preserves and is preserved by semantic compositionality (composition of moduli of continuity).

The main contributions of this chapter are:

1. We develop for each of the compositionality properties of non-extensiveness, non-expansiveness, Lipschitz continuity and uniform continuity appropriate SOS rule

and specification formats guaranteeing that the specified operators satisfy the respective compositionality property (Definitions 4.4,4.18,4.28,4.37).

2. We show for each SOS rule and specification format by appropriate examples that our syntactic rule constraints cannot be relaxed in any obvious way.

3. We analyze the SOS specifications of several process algebra operators and determine their respective compositionality properties (Corollaries 4.10,4.23,4.25).

4. We apply those results and derive an upper bound on the distance between two language expressions by inspecting solely the syntactic compositionality properties of the operators used in the expressions (Propositions 4.12,4.26,4.40 and Theorem 4.61).

5. We provide a method that allows us to derive for any uniformly continuous operator its respective modulus of continuity (i.e. its compositionality property) from its specification rules (Theorems 4.53 and 4.54).

6. We provide a method that given any modulus of continuity determines sufficient syntactic requirements s.t. any specification satisfying these requirements defines an operator with that modulus of continuity (Theorem 4.58).

7. We identify sufficient properties of any behavioral metric ensuring that the compositionality results (modulus of continuity of the specified operators and the distance bounds between language expressions) are applicable also to this behavioral metric (Theorem 4.73).

This chapter has been partially published as [GT15].

4.2 Non-extensive operators

We start in this section with non-extensiveness as the strictest compositionality property and proceed in the subsequent sections with the weaker compositionality properties of non-expansiveness, Lipschitz continuity and uniform continuity. We will start by introducing formally the compositionality property then propose an appropriate rule and specification format guaranteeing that all specified operators satisfy the considered compositionality property, proceed by exploring which process algebra operators satisfy the considered compositionality property, and conclude with a simple method to compute an upper bound on the distance between closed instances of terms by inspecting solely the compositionality properties of their operators. The specification formats for the weaker compositionality properties build on top of the specification formats for the stricter compositionality properties in order to allow for expressive specification formats.

The compositionality property of q-non-extensiveness bases on the q-norm. In the former chapter we focussed only on ∞-non-extensiveness (Definition 3.4) and will generalize now to q-non-extensiveness for arbitrary $q \in [1, \infty]$.

$$\frac{}{a.\bigoplus_{i=1}^{n}[p_i]x_i \xrightarrow{a} \sum_{i=1}^{n} p_i\delta(x_i)} \qquad \frac{x \xrightarrow{a} \mu}{x+y \xrightarrow{a} \mu} \qquad \frac{y \xrightarrow{a} \nu}{x+y \xrightarrow{a} \nu}$$

$$\frac{x \xrightarrow{a} \mu \quad y \not\xrightarrow{a}}{x +_p y \xrightarrow{a} \mu} \qquad \frac{x \not\xrightarrow{a} \quad y \xrightarrow{a} \nu}{x +_p y \xrightarrow{a} \nu} \qquad \frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu}{x +_p y \xrightarrow{a} \mu \oplus_p \nu}$$

Figure 4.2: Non-extensive process algebra operators

Definition 4.1 (Non-extensive operator)**.** Let $P = (\Sigma, A, R)$ be a PTSS. Given $q \in [1, \infty]$ we say that an operator $f \in \Sigma$ is *q-non-extensive* w.r.t. λ-bisimilarity metric $\mathbf{d}$ if

$$\mathbf{d}(f(s_1,\ldots,s_n), f(t_1,\ldots,t_n)) \leq \begin{cases} \left(\sum_{i=1}^{n} \mathbf{d}(s_i,t_i)^q\right)^{\frac{1}{q}} & \text{if } q \in [1,\infty) \\ \max_{i=1}^{n} \mathbf{d}(s_i,t_i) & \text{if } q = \infty \end{cases}$$

for all closed terms $s_i, t_i \in \mathsf{T}(\Sigma)$. We call f *non-extensive* w.r.t. $\mathbf{d}$ if f is ∞-non-extensive w.r.t. $\mathbf{d}$.

If an operator f is q-non-extensive for some $q \in [1, \infty]$, then f is also q'-non-extensive for all $1 \leq q' \leq q$. In other words, the compositionality property of q-non-extensiveness gets weaker if q decreases. In this section we focus on ∞-non-extensiveness and consider q-non-extensiveness for $q \in [1, \infty)$ in Section 4.4.

4.2.1 Analysis of non-extensive operators

We discuss first a few examples that show which rule patterns specify and which rule patterns do not specify non-extensive operators. We start by analyzing how many times a source process or its derivatives may appear in the rule target. Intuitively, multiple occurrences of some source process or its derivatives in the rule target may be understood as replication of this process along the transition specified by that rule. We start with analyzing unary operators.

Example 4.2. Consider the rules

$$\frac{x \xrightarrow{a} \mu}{f(x) \xrightarrow{a} \theta} \qquad \frac{x \xrightarrow{a} \mu}{x+y \xrightarrow{a} \mu} \qquad \frac{y \xrightarrow{a} \nu}{x+y \xrightarrow{a} \nu}$$

with $\theta \in \mathbb{DT}(\Sigma)$ any open distribution term. For the discussion in this example we assume that the nondeterministic alternative composition operator $+$ is non-extensive (formally shown below in Corollary 4.10). In the following we analyze in which cases the specified operator f is non-extensive by considering various distribution terms θ. As arguments for operator f we will use the closed terms $s = a.b.0$ and $t = a.([1-\epsilon]b.0 \oplus [\epsilon]c.0)$ with ϵ some fixed value in $(0,1)$. The operators $a.([p_1]\_ \oplus \ldots \oplus [p_n]\_)$ (probabilistic prefix operator $a.\bigoplus_{i=1}^{n}[p_i]\_$) and constant 0 (stop process) are specified by the rules

given in Figure 4.2. For brevity, we write $a.\_$ for $a.([1]\_)$. The SOS rules allow us to derive the transitions $s \xrightarrow{a} \delta(b.0)$ and $t \xrightarrow{a} (1-\epsilon)\delta(b.0) + \epsilon\delta(c.0)$. Hence $\mathbf{d}(s,t) = \lambda \cdot \mathbf{K}(\mathbf{d})(\delta(b.0), (1-\epsilon)\delta(b.0) + \epsilon\delta(c.0)) = \lambda \cdot ((1-\epsilon)\cdot 0 + \epsilon \cdot 1) = \lambda \cdot \epsilon$.

Consider $\theta = \delta(x) + \delta(x)$. The source process x appears twice in the rule target in the context of the alternative composition operator. Intuitively, the source process x gets replicated, but the two instances of x do not evolve in the specified transition. The transitions $f(s) \xrightarrow{a} \delta(s) + \delta(s)$ and $f(t) \xrightarrow{a} \delta(t) + \delta(t)$ are derivable. We get $\mathbf{d}(f(s), f(t)) = \lambda \cdot \mathbf{K}(\mathbf{d})(\delta(s)+\delta(s), \delta(t)+\delta(t)) = \lambda \cdot \mathbf{d}(s+s, t+t) = \lambda \cdot \mathbf{d}(s,t) \leq \mathbf{d}(s,t)$. The non-extensiveness condition is satisfied for these specific arguments s and t. Theorem 4.5 below will confirm that this specification of operator f is non-extensive.

Consider $\theta = \mu + \mu$. The x-derivative μ appears twice in the rule target θ in the context of the alternative composition operator. Intuitively, the source process x evolves and gets replicated in the specified transition. The transitions $f(s) \xrightarrow{a} \pi_s$, with $\pi_s = \delta(b.0 + b.0)$, and $f(t) \xrightarrow{a} \pi_t$, with $\pi_t = (1-\epsilon)^2\delta(b.0 + b.0) + \epsilon(1-\epsilon)\delta(b.0 + c.0) + \epsilon(1-\epsilon)\delta(c.0 + b.0) + \epsilon^2\delta(c.0 + c.0)$, are derivable. Now, $\mathbf{K}(\mathbf{d})(\pi_s, \pi_t) = 1 - (1-\epsilon)^2$. The terms s and t are witnesses for the violation of the non-extensiveness condition $\mathbf{d}(f(s), f(t)) = \lambda \cdot \mathbf{K}(\mathbf{d})(\pi_s, \pi_t) = \lambda \cdot (1-(1-\epsilon)^2) > \lambda \cdot \epsilon = \mathbf{d}(s,t)$. Hence, this specification of operator f is not non-extensive.

Consider $\theta = (\mu + \mu) \oplus_r \delta(0)$ for some fixed $r \in (0,1)$. Intuitively, the source process x evolves and gets replicated with probability r, while x terminates with the remaining probability $1-r$. The transitions $f(s) \xrightarrow{a} \pi'_s = r\pi_s + (1-r)\delta(0)$ and $f(t) \xrightarrow{a} \pi'_t = r\pi_t + (1-r)\delta(0)$ are derivable, with π_s and π_t from the former case $\theta = \mu + \mu$. The distributions π'_s and π'_t have a distance $\mathbf{K}(\mathbf{d})(\pi'_s, \pi'_t) = r(1-(1-\epsilon)^2)$. Hence, for $r \leq 0.5$ we get $\mathbf{d}(f(s), f(t)) = \lambda \cdot \mathbf{K}(\mathbf{d})(\pi'_s, \pi'_t) = \lambda r \cdot (1-(1-\epsilon)^2) \leq \lambda r \cdot 2\epsilon \leq \lambda \cdot \epsilon = \mathbf{d}(s,t)$ and the non-extensiveness condition is satisfied for these specific arguments s and t. Theorem 4.5 below will confirm that for $r \leq 0.5$ this specification of operator f is non-extensive.

Consider $\theta = (\delta(x) + \mu) \oplus_r \delta(0)$ for some fixed $r \in (0,1)$. Intuitively, with probability r the source process x gets replicated but only one of the two instances evolves, while with the remaining probability $1-r$ the source process x terminates. The transitions $f(s) \xrightarrow{a} \pi_s$, with $\pi_s = r\delta(s + b.0) + (1-r)\delta(0)$, and $f(t) \xrightarrow{a} \pi_t$, with $\pi_t = r((1-\epsilon)\delta(t + b.0) + \epsilon\delta(t + c.0)) + (1-r)\delta(0)$, are derivable. Now, $\mathbf{K}(\mathbf{d})(\pi_s, \pi_t) = r((1-\epsilon)\mathbf{d}(s,t) + \epsilon)$. For these specific arguments s and t the non-extensiveness condition is satisfied if $\lambda\, \mathbf{K}(\mathbf{d})(\pi_s, \pi_t) = \lambda r((1-\epsilon)\lambda\epsilon + \epsilon) \leq \lambda\epsilon$, i.e. if $r \leq 1/(1 + \lambda(1-\epsilon))$. Since $1/(1 + \lambda(1-\epsilon)) \geq 1/(1+\lambda)$, the non-extensiveness condition is satisfied if $r \leq 1/(1+\lambda)$. Note that this expression is independent of the bisimulation distance between s and t. Theorem 4.5 below will confirm that for $r \leq 1/(1+\lambda)$ this specification of operator f is non-extensive.

In essence, Example 4.2 shows that non-extensive operators may copy processes that do not evolve in the specified transition (technically, multiple instances of the source process may occur in the rule target). Furthermore, if a process evolves, then the number of evolved instances weighted by the probability of their realization may not exceed 1 (technically, the number of occurrences of derivatives in the rule target weighted by the probability of their respective convex combination contexts has to be at most 1).

We proceed now by analyzing operators with multiple arguments and investigate how many instances of each source process may get delayed or may evolve.

Example 4.3. Consider the rules

$$\frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu}{f(x,y) \xrightarrow{a} \theta} \qquad \frac{x \xrightarrow{a} \mu}{x + y \xrightarrow{a} \mu} \qquad \frac{y \xrightarrow{a} \nu}{x + y \xrightarrow{a} \nu}$$

with $\theta \in \mathbb{DT}(\Sigma)$ any open distribution term. Recall that the operator $+$ is non-extensive. We consider again various distribution terms θ and analyze in which cases the specified operator f is non-extensive. As arguments for operator f we will use the pairs of terms $s_1 = a.b_1.0$ and $s_2 = a.b_2.0$, and $t_1 = a.([1-\epsilon_1]b_1.0 \oplus [\epsilon_1]c_1.0)$ and $t_2 = a.([1-\epsilon_2]b_2.0 \oplus [\epsilon_2]c_2.0)$ with some fixed $\epsilon_1, \epsilon_2 \in (0,1)$. Clearly, $\mathbf{d}(s_1,t_1) = \lambda \cdot \epsilon_1$ and $\mathbf{d}(s_2,t_2) = \lambda \cdot \epsilon_2$.

Consider $\theta = (\delta(x) + \delta(y)) + (\delta(x) + \delta(y))$. The source process x and the source process y each appear twice in the rule target in the context of the non-extensive alternative composition operator. Intuitively, the source processes x and y both get replicated but do not evolve in the specified transition. The transitions $f(s_1,s_2) \xrightarrow{a} \pi_s$, with $\pi_s = \delta((s_1 + s_2) + (s_1 + s_2))$, and $f(t_1,t_2) \xrightarrow{a} \pi_t$, with $\pi_t = \delta((t_1 + t_2) + (t_1 + t_2))$, are derivable. We get $\mathbf{d}(f(s_1,s_2), f(t_1,t_2)) = \lambda \cdot \mathbf{K}(\mathbf{d})(\pi_s,\pi_t) = \lambda \cdot \mathbf{K}(\mathbf{d})(\delta((s_1+s_2)+(s_1+s_2)), \delta((t_1+t_2)+(t_1+t_2))) \leq \lambda \cdot \max(\mathbf{d}(s_1,t_1), \mathbf{d}(s_2,t_2)) \leq \max(\mathbf{d}(s_1,t_1), \mathbf{d}(s_2,t_2))$. The non-extensiveness condition is satisfied for these specific pairs of arguments (s_1,s_2) and (t_1,t_2). Theorem 4.5 below will confirm that this specification of operator f is non-extensive.

Consider $\theta = \mu + \nu$. Both the x-derivative μ and the y-derivative ν appear once in the rule target θ in the context of the non-extensive alternative composition operator. Intuitively, both source processes x and y evolve in the specified transition. The transitions $f(s_1,s_2) \xrightarrow{a} \pi_s$, with $\pi_s = \delta(b_1.0 + b_2.0)$, and $f(t_1,t_2) \xrightarrow{a} \pi_t$, with $\pi_t = (1-\epsilon_1)(1-\epsilon_2)\delta(b_1.0+b_2.0) + \epsilon_1(1-\epsilon_2)\delta(c_1.0+b_2.0) + (1-\epsilon_1)\epsilon_2\delta(b_1.0+c_2.0) + \epsilon_1\epsilon_2\delta(c_1.0+c_2.0)$, are derivable. Now, $\mathbf{K}(\mathbf{d})(\pi_s,\pi_t) = 1-(1-\epsilon_1)(1-\epsilon_2)$. The pairs of terms (s_1,s_2) and (t_1,t_2) are witnesses for the violation of the non-extensiveness condition $\mathbf{d}(f(s_1,s_2), f(t_1,t_2)) = \lambda \cdot \mathbf{K}(\mathbf{d})(\pi_s,\pi_t) = \lambda(1-(1-\epsilon_1)(1-\epsilon_2)) > \max(\lambda \cdot \epsilon_1, \lambda \cdot \epsilon_2) = \max(\mathbf{d}(s_1,t_1), \mathbf{d}(s_2,t_2))$. Hence, this specification of operator f is not non-extensive.

Consider $\theta = (\mu + \nu) \oplus_r \delta(0)$ for some fixed $r \in (0,1)$. Intuitively, with probability r both source processes x and y evolve in the specified transition, while they terminate with the remaining probability $1-r$. The distributions accessible from $f(s_1,s_2)$ and $f(t_1,t_2)$ are $\pi'_s = r\pi_s + (1-r)\delta(0)$ and $\pi'_t = r\pi_t + (1-r)\delta(0)$, resp. , with π_s and π_t from the former case $\theta = \mu + \nu$, and have a distance $\mathbf{K}(\mathbf{d})(\pi'_s,\pi'_t) = r \cdot \mathbf{K}(\mathbf{d})(\pi_s,\pi_t) = r \cdot (1-(1-\epsilon_1)(1-\epsilon_2)) \leq r \cdot (\epsilon_1 + \epsilon_2)$. Hence, $\mathbf{d}(f(s_1,s_2), f(t_1,t_2)) = \lambda \cdot \mathbf{K}(\mathbf{d})(\pi'_s,\pi'_t) \leq \lambda \cdot r \cdot (\epsilon_1 + \epsilon_2) = r \cdot (\mathbf{d}(s_1,t_1) + \mathbf{d}(s_2,t_2))$. For these specific pairs of arguments (s_1,s_2) and (t_1,t_2) the non-extensiveness condition $r \cdot (\mathbf{d}(s_1,t_1) + \mathbf{d}(s_2,t_2)) \leq \max(\mathbf{d}(s_1,t_2), \mathbf{d}(s_2,t_2))$ is satisfied if $r \leq 0.5$. Theorem 4.5 below will confirm that for $r \leq 0.5$ this specification of operator f is non-extensive.

Consider $\theta = (\delta(x) + \nu) \oplus_r \delta(0)$ for some fixed $r \in (0,1)$. Intuitively, with probability r the source process x gets delayed by one step and the source process y evolves in the specified transition, while both processes terminate with the remaining probability $1-r$. The transitions $f(s_1,s_2) \xrightarrow{a} \pi_s$, with $\pi_s = r\delta(s_1 + b_2.0) + (1-r)\delta(0)$, and $f(t_1,t_2) \xrightarrow{a} \pi_t$, with $\pi_t = r((1-\epsilon_2)\delta(t_1 + b_2.0) + \epsilon_2\delta(t_1 + c_2.0)) + (1-r)\delta(0)$ are derivable. Now, $\mathbf{K}(\mathbf{d})(\pi_s,\pi_t) = r((1-\epsilon_2)\mathbf{d}(s_1,t_1) + \epsilon_2)$. For these specific pairs of arguments (s_1,s_2) and (t_1,t_2) the non-extensiveness condition $\mathbf{d}(f(s_1,s_2), f(t_1,t_2)) =$

$\lambda \cdot \mathbf{K}(\mathbf{d})(\pi_s, \pi_t) = \lambda \cdot r \cdot ((1-\epsilon_2)\mathbf{d}(s_1,t_1)+\epsilon_2) \leq \max(\mathbf{d}(s_1,t_1),\mathbf{d}(s_2,t_2))$ is satisfied if $r \leq \max(\mathbf{d}(s_1,t_1),\mathbf{d}(s_2,t_2))/(\lambda((1-\epsilon_2)\mathbf{d}(s_1,t_1)+\epsilon_2)) = \max(\epsilon_1,\epsilon_2)/((1-\epsilon_2)\lambda_1\epsilon_1+\epsilon_2)$. Theorem 4.5 below will confirm that for $r \leq \max(\epsilon_1,\epsilon_2)/((1-\epsilon_2)\lambda_1\epsilon_1+\epsilon_2)$ this specification of operator f is non-extensive.

In essence, Example 4.3 shows that non-extensive operators may copy all argument processes that do not evolve in the specified transition (technically, multiple instances of each of the source processes may occur in the rule target). Furthermore, if processes evolve, then the number of instances of all evolved processes together weighted by the probability of their realization may not exceed 1 (technically, the number of all occurrences of all derivatives in the rule target weighted by the probability of their respective convex combination contexts has to be be at most 1).

4.2.2 Specification of non-extensive operators

The compositionality properties of non-extensiveness, non-expansiveness, Lipschitz continuity and uniform continuity require that the respective operators copy their arguments only a limited number of times. To formalize these requirements we introduce the mapping $\mathrm{Var}\colon (\mathcal{V} \times (\mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma))) \to \mathbb{R}_{\geq 0}$ that gives for any state or distribution term the weighted number of occurrences of its variables (cf. analysis in Examples 4.2 and 4.3). Applying Var to the target of rules will allow us to formulate for each compositionality property an expressive sufficient condition to verify if the specified operator satisfies the compositionality property.

Since the number of occurrences of some source variable $x \in \mathcal{V}_s$ in the rule target does not affect non-extensiveness of the specified operator, we define $\mathrm{Var}(x, f(t_1,\ldots,t_n))$ (resp. $\mathrm{Var}(x, f(\theta_1,\ldots,\theta_n))$) as the maximum over the values $\mathrm{Var}(x,t_i)$ (resp. $\mathrm{Var}(x,\theta_i)$). On the contrary, since non-extensiveness of the specified operator depends on how many times the derivatives $\mu \in \mathcal{V}_d$ of source variables appear in the rule target, we define $\mathrm{Var}(\mu, f(\theta_1,\ldots,\theta_n))$ by summing up the values $\mathrm{Var}(\mu,\theta_i)$. Finally, the number of occurrences of variables in a distribution term θ are weighted by the probabilistic choices in the convex combinations occurring in θ. Formally:

$$\mathrm{Var}(\zeta,t) = \begin{cases} 1 & \text{if } t = \zeta \\ \max\limits_{i=1}^{n} \mathrm{Var}(\zeta,t_i) & \text{if } t = f(t_1,\ldots,t_n) \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{Var}(\zeta,\theta) = \begin{cases} 1 & \text{if } \theta = \zeta \\ \mathrm{Var}(\zeta,t) & \text{if } \theta = \delta(t) \\ \sum\limits_{i\in I} p_i \cdot \mathrm{Var}(\zeta,\theta_i) & \text{if } \theta = \sum\limits_{i\in I} p_i\theta_i \\ \max\limits_{i=1}^{n} \mathrm{Var}(\zeta,\theta_i) & \text{if } \theta = f(\theta_1,\ldots,\theta_n) \wedge \zeta \in \mathcal{V}_s \\ \sum\limits_{i=1}^{n} \mathrm{Var}(\zeta,\theta_i) & \text{if } \theta = f(\theta_1,\ldots,\theta_n) \wedge \zeta \in \mathcal{V}_d \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

We proceed by formalizing the copying of processes in rules (cf. analysis in Examples 4.2 and 4.3). We define the mapping $\mathsf{copy}\colon R \times (0,1] \to \mathbb{R}_{\geq 0}$ to characterize for each rule $r \in R$ how many instances of the source processes are delayed (technically, the number of occurrences of the source variables in the rule target) and how many instances of the source processes evolve (technically, the number of occurrences of derivatives in the rule target). Since source variables x_i may appear multiple times in the target of a rule r without compromising non-extensiveness, we consider the maximum of $\mathrm{Var}(x_i, \mathrm{trgt}(r))$ over all x_i. Moreover, since source variables that appear in the rule target represent processes that are delayed by one step, we need to discount them by λ. On the contrary, since non-extensiveness of the specified operator depends on how many times derivatives appear in the rule target, we consider the sum of $\mathrm{Var}(\mu, \mathrm{trgt}(r))$ over all derivatives μ. Formally:

$$\mathsf{copy}(r,\lambda) = \lambda \cdot \max_{i \in \{1,\ldots,n\}} \mathrm{Var}(x_i, \mathrm{trgt}(r)) + \sum_{\substack{i \in \{1,\ldots,n\} \\ \mu \in \mathrm{der}(r,x_i)}} \mathrm{Var}(\mu, \mathrm{trgt}(r)). \tag{4.2}$$

A PGSOS rule specifies a non-extensive operator if the number of (evolved and delayed) source process instances does not exceed 1.

Definition 4.4 (Non-extensiveness format)**.** A PGSOS rule r is a *λ-non-extensive rule* if

$$\mathsf{copy}(r,\lambda) \leq 1.$$

A PTSS (Σ, A, R) is a *λ-non-extensive PTSS* if all rules $r \in R$ are λ-non-extensive rules.

Non-extensive PTSSs specify non-extensive operators.

Theorem 4.5. *Let $P = (\Sigma, A, R)$ be a λ-non-extensive PTSS. Then all operators $f \in \Sigma$ are non-extensive w.r.t. λ-bisimilarity metric* $\mathbf{d}$.

To prove Theorem 4.5 we define the congruence closure of a pseudometric as the quantitative analogous to the well-known congruence closure of an equivalence. The congruence closure of a 1-bounded pseudometric d w.r.t. a Σ-indexed family of moduli of continuity[1] $(m)_{f \in \Sigma}$ is the largest function $\mathsf{cl}_m(d)\colon \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \to [0,1]$ with $\mathsf{cl}_m(d) \sqsubseteq d$ such that each operator $f \in \Sigma$ satisfies the respective modulus of continuity m_f. To prove Theorem 4.5 we show then that the congruence closure $\mathsf{cl}_m(\mathbf{d})$ of the bisimilarity metric $\mathbf{d}$ is a prefixed point of $\mathbf{B}$ on $([0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)}, \sqsubseteq)$, i.e. $\mathbf{B}(\mathsf{cl}_m(\mathbf{d})) \sqsubseteq \mathsf{cl}_m(\mathbf{d})$. From $\mathsf{cl}_m(\mathbf{d}) \sqsubseteq \mathbf{d}$ and the fact that $\mathbf{d}$ is the least prefixed point of $\mathbf{B}$ on $([0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)}, \sqsubseteq)$ we conclude then $\mathsf{cl}_m(\mathbf{d}) = \mathbf{d}$. Hence, each operator $f \in \Sigma$ satisfies the modulus of continuity m_f w.r.t. $\mathbf{d}$.

Definition 4.6 (Congruence closure)**.** Assume any signature Σ and a Σ-indexed family of moduli of continuity $(m)_{f\in\Sigma}$ with $m_f : [0,1]^n \to [0,1]$. The *congruence closure* of some pseudometric $d\colon \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \to [0,1]$ w.r.t. $(m)_{f\in\Sigma}$ is defined as the function $\mathsf{cl}_m(d)\colon \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \to [0,1]$ with

$$\mathsf{cl}_m(d)(s,t) = \begin{cases} \min(d(s,t), m_f(\mathsf{cl}_m(d)(s_1,t_1), \ldots, \mathsf{cl}_m(d)(s_n,t_n))) & \text{if } \begin{bmatrix} s = f(s_1,\ldots,s_n) \\ t = f(t_1,\ldots,t_n) \end{bmatrix} \\ d(s,t) & \text{otherwise} \end{cases}$$

for all $s, t \in \mathsf{T}(\Sigma)$.

[1] The notion of modulus of continuity is standard and recalled in Definition 4.30.

By induction over the minimum depth of s and t it can be shown that the congruence closure is well-defined (existence and uniqueness). By the construction of $\mathsf{cl}_m(d)$ it is clear that $\mathsf{cl}_m(d) \sqsubseteq d$ and all operators $f \in \Sigma$ satisfy the respective modulus of continuity m_f, i.e. $\mathsf{cl}_m(d)(f(s_1,\dots,s_n), f(t_1,\dots,t_n)) \leq m_f(\mathsf{cl}_m(d)(s_1,t_1),\dots,\mathsf{cl}_m(d)(s_n,t_n))$.

Definition 4.7 (Non-extensive congruence closure)**.** Let $d\colon \mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)\to[0,1]$ be any function. We call $\mathsf{cl}_m(d)$ with $(m)_{f\in\Sigma}$ defined by

$$m_f(\epsilon_1,\dots,\epsilon_n) = \max_{i=1}^{n} \epsilon_i$$

for all $f \in \Sigma$ the *non-extensive congruence closure* of d w.r.t. Σ.

The following two lemmas give an upper bound on the distance between closed instances of state and distribution terms built of non-extensive operators.

Lemma 4.8. *Let Σ be any signature, $d_1\colon T(\Sigma)\times T(\Sigma)\to[0,1]$ any function, and d_2 the non-extensive congruence closure of d_1 w.r.t. Σ. Then, for any state term $t \in \mathbb{T}(\Sigma)$ we have*

$$d_2(\sigma(t),\sigma'(t)) \leq \max_{x\in\mathcal{V}_s} d_2(\sigma(x),\sigma'(x))\cdot \mathrm{Var}(x,t)$$

for all closed substitutions $\sigma,\sigma'\colon \mathcal{V}_s \to T(\Sigma)$.

Proof. By structural induction over t. The base case $t = x$ with $x \in \mathcal{V}_s$ follows immediately from $\mathrm{Var}(x,x) = 1$. Consider the induction step $t = f(t_1,\dots,t_n)$. Then we have

$$\begin{aligned}
&d_2(\sigma(t),\sigma'(t))\\
&\leq \max_{i=1,\dots,n} d_2(\sigma(t_i),\sigma'(t_i)) && \text{(definition of } d_2\text{)}\\
&\leq \max_{i=1,\dots,n}\max_{x\in\mathcal{V}_s} d_2(\sigma(x),\sigma'(x))\cdot\mathrm{Var}(x,t_i) && \text{(inductive hypothesis)}\\
&= \max_{x\in\mathcal{V}_s}\max_{i=1,\dots,n} d_2(\sigma(x),\sigma'(x))\cdot\mathrm{Var}(x,t_i)\\
&= \max_{x\in\mathcal{V}_s} d_2(\sigma(x),\sigma'(x))\cdot\mathrm{Var}(x,t) && \text{(definition of Var).}
\end{aligned}$$

□

Lemma 4.9. *Let Σ be any signature, $d_1\colon T(\Sigma)\times T(\Sigma)\to[0,1]$ any function, and d_2 the non-extensive congruence closure of d_1 w.r.t. Σ. Then, for any distribution term $\theta \in \mathbb{DT}(\Sigma)$ we have*

$$\mathbf{K}(d_2)(\sigma(\theta),\sigma'(\theta)) \leq \max_{x\in\mathcal{V}_s} d_2(\sigma(x),\sigma'(x))\cdot\mathrm{Var}(x,\theta) + \sum_{\mu\in\mathcal{V}_d} \mathbf{K}(d_2)(\sigma(\mu),\sigma'(\mu))\cdot\mathrm{Var}(\mu,\theta)$$

for all closed substitutions $\sigma,\sigma'\colon \mathcal{V}\to T(\Sigma)\cup DT(\Sigma)$.

Proof. Without loss of generality, assume that θ is in normal form (Definition 2.8 and Proposition 2.9). By induction over θ. The base case $\theta = \mu$ follows immediately from $\mathrm{Var}(\mu,\mu) = 1$. For $\theta = \delta(x)$ the thesis follows directly from $\mathbf{K}(d_2)(\sigma(\delta(x)),\sigma'(\delta(x))) \leq d_2(\sigma(x),\sigma'(x))$ (Proposition 2.32.2), $d_2(\sigma(x),\sigma'(x)) \leq \max_{x\in\mathcal{V}_s} d_2(\sigma(x),\sigma'(x))\cdot\mathrm{Var}(x,x)$ (Lemma 4.8) and $\mathrm{Var}(x,x) = \mathrm{Var}(x,\delta(x))$.

Consider the induction step $\theta = f(\theta_1, \ldots, \theta_n)$. Remind that none of the distribution terms θ_i contains any convex combination since θ is in normal form. We will prove the thesis in two steps. First, we build a suitable matching $\omega_\theta \in \Omega(\sigma(\theta), \sigma'(\theta))$ satisfying

$$\mathbf{K}(d_2)(\sigma(\theta), \sigma'(\theta)) \leq \sum_{t,t' \in \mathsf{T}(\Sigma)} \omega_\theta(t, t') \cdot d_2(t, t'). \tag{4.3}$$

Subsequently, we show that the matching ω_θ satisfies the thesis as follows

$$\begin{aligned} &\sum_{t,t' \in \mathsf{T}(\Sigma)} \omega_\theta(t, t') \cdot d_2(t, t') \leq \\ &\qquad \max_{x \in \mathscr{V}_s} d_2(\sigma(x), \sigma'(x)) \cdot \mathsf{Var}(x, \theta) + \sum_{\mu \in \mathscr{V}_d} \mathbf{K}(d_2)(\sigma(\mu), \sigma'(\mu)) \cdot \mathsf{Var}(\mu, \theta). \end{aligned} \tag{4.4}$$

To construct the matching ω_θ, we define first a matching ω_ζ for all variables $\zeta \in \mathsf{Var}(\theta)$. For any $x \in \mathscr{V}_s \cap \mathsf{Var}(\theta)$, let $\omega_x \in \Omega(\delta(\sigma(x)), \delta(\sigma'(x)))$ be the unique matching between $\sigma(\delta(x))$ and $\sigma'(\delta(x))$ defined by $\omega_x(\sigma(x), \sigma'(x)) = 1$. For any $\mu \in \mathscr{V}_d \cap \mathsf{Var}(\theta)$, let $\omega_\mu \in \Omega(\sigma(\mu), \sigma'(\mu))$ be any of the optimal (possibly not unique) matchings between $\sigma(\mu)$ and $\sigma'(\mu)$ such that $\mathbf{K}(d_2)(\sigma(\mu), \sigma'(\mu)) = \sum_{t,t' \in \mathsf{T}(\Sigma)} \omega_\mu(t, t') \cdot d_2(t, t')$.

Then, let ω_θ be defined by $\omega_{g(\theta_1,\ldots,\theta_n)}(g(t_1, \ldots, t_n), g(t'_1, \ldots, t'_n)) = \prod_{i=1}^n \omega_{\theta_i}(t_i, t'_i)$. We show that ω_θ is a matching between $\sigma(\theta)$ and $\sigma'(\theta)$, i.e. $\omega \in \Omega(\sigma(\theta), \sigma'(\theta))$. We proceed by inductively showing that $\omega_{g(\theta_1,\ldots,\theta_n)}$ is a matching between $\sigma(g(\theta_1, \ldots, \theta_n))$ and $\sigma'(g(\theta_1, \ldots, \theta_n))$, i.e. $\omega_{g(\theta_1,\ldots,\theta_n)} \in \Omega(\sigma(g(\theta_1, \ldots, \theta_n)), \sigma'(g(\theta_1, \ldots, \theta_n)))$, by assuming that ω_{θ_i} is a matching for $\sigma(\theta_i)$ and $\sigma'(\theta_i)$, i.e. $\omega_{\theta_i} \in \Omega(\sigma(\theta_i), \sigma'(\theta_i))$. In detail, we show that the left marginal of $\omega_{g(\theta_1,\ldots,\theta_n)}$ is $\sigma(g(\theta_1, \ldots, \theta_n))$, the proof that the right marginal is $\sigma'(g(\theta_1, \ldots, \theta_n))$ is analogous. Let $t = g(t_1, \ldots, t_n)$. Recall that $\omega_{\theta_i} \in \Omega(\sigma(\theta_i), \sigma'(\theta_i))$ implies that $\sum_{t'_i \in \mathsf{T}(\Sigma)} \omega_{\theta_i}(t_i, t'_i) = \sigma(\theta_i)(t_i)$. Then we have

$$\begin{aligned} &\sum_{t' \in \mathsf{T}(\Sigma)} \omega_{g(\theta_1,\ldots,\theta_n)}(t, t') \\ =&\sum_{t'_1,\ldots,t'_n \in \mathsf{T}(\Sigma)} \omega_{g(\theta_1,\ldots,\theta_n)}(g(t_1, \ldots, t_n), g(t'_1, \ldots, t'_n)) \\ =&\sum_{t'_1,\ldots,t'_n \in \mathsf{T}(\Sigma)} \prod_{i=1}^n \omega_{\theta_i}(t_i, t'_i) \\ =&\prod_{i=1}^n \sum_{t'_i \in \mathsf{T}(\Sigma)} \omega_{\theta_i}(t_i, t'_i) \\ =&\prod_{i=1}^n \sigma(\theta_i)(t_i) \\ =&\sigma(g(\theta_1, \ldots, \theta_n))(t) \end{aligned}$$

whereby the distribution of the summation over the product from step 3 to 4 can be shown by induction as in the proof of Theorem 2.33. Hence we conclude that ω_θ is a matching satisfying Equation 4.3.

Now we construct for the distribution term θ a state term t_θ s.t. all terms in the support of $\sigma(\theta)$ and $\sigma'(\theta)$ are instances of t_θ. Let k_μ be the number of occurrences of the distribution variable μ in θ. We define t_θ as that state term derived from θ by replacing $\delta(x)$ with x and by replacing each distribution variable μ with some fresh variable $y_{(\mu,j)} \notin \mathsf{Var}(\theta)$ such that each $y_{(\mu,j)}$ (with $j = 1,\ldots,k_\mu$) occurs only once in t_θ. Clearly, for any closed substitution σ_1 with $\sigma_1(x) = \sigma(x)$ we have

$$\sigma(\theta)(\sigma_1(t_\theta)) = \prod_{\mu\in\mathsf{Var}(\theta)} \prod_{j=1}^{k_\mu} \sigma(\mu)(\sigma_1(y_{\mu,j})).$$

Then, for any closed substitutions σ_1, σ_1' with $\sigma_1(x) = \sigma(x)$ and $\sigma_1'(x) = \sigma'(x)$ for all $x \in \mathsf{Var}(t)$, the matching ω_θ is

$$\omega_\theta(\sigma_1(t_\theta), \sigma_1'(t_\theta)) = \left(\prod_{x\in\mathsf{Var}(\theta)} \omega_x(\sigma_1(x), \sigma_1'(x))\right) \cdot \left(\prod_{\substack{\mu\in\mathbf{Var}(\theta)\\ j=1,\ldots,k_\mu}} \omega_\mu(\sigma_1(y_{(\mu,j)}), \sigma_1'(y_{(\mu,j)}))\right).$$

It remains to show Equation 4.4. We have

$$\begin{aligned}
&\sum_{t,t'\in\mathsf{T}(\Sigma)} \omega_\theta(t,t') \cdot d_2(t,t') \\
&= \sum_{\sigma_1,\sigma_1'\in\mathsf{T}(\Sigma)^{\mathcal{V}_s}} \omega_\theta(\sigma_1(t_\theta), \sigma_1'(t_\theta)) \cdot d_2(\sigma_1(t_\theta), \sigma_1'(t_\theta)) \\
&= \sum_{\sigma_1,\sigma_1'\in\mathsf{T}(\Sigma)^{\mathcal{V}_s}} \left(\left(\prod_{x\in\mathsf{Var}(\theta)} \omega_x(\sigma_1(x), \sigma_1'(x))\right) \cdot \left(\prod_{\substack{\mu\in\mathbf{Var}(\theta)\\ j=1,\ldots,k_\mu}} \omega_\mu(\sigma_1(y_{(\mu,j)}), \sigma_1'(y_{(\mu,j)}))\right)\right) \cdot \\
&\qquad d_2(\sigma_1(t_\theta), \sigma_1'(t_\theta)) \\
&\leq \sum_{\sigma_1,\sigma_1'\in\mathsf{T}(\Sigma)^{\mathcal{V}_s}} \left(\left(\prod_{x\in\mathsf{Var}(\theta)} \omega_x(\sigma_1(x), \sigma_1'(x))\right) \cdot \left(\prod_{\substack{\mu\in\mathbf{Var}(\theta)\\ j=1,\ldots,k_\mu}} \omega_\mu(\sigma_1(y_{(\mu,j)}), \sigma_1'(y_{(\mu,j)}))\right)\right) \cdot \\
&\qquad \max\left(\max_{x\in\mathsf{Var}(\theta)} d_2(\sigma_1(x), \sigma_1'(x)), \max_{\substack{\mu\in\mathbf{Var}(\theta)\\ j=1,\ldots,k_\mu}} d_2(\sigma_1(y_{(\mu,j)}), \sigma_1'(y_{(\mu,j)}))\right) \\
&\quad \text{(by Lemma 4.8)} \\
&= \sum_{\substack{\sigma_1,\sigma_1'\in\mathsf{T}(\Sigma)^{\mathcal{V}_s}\\ \sigma_1(x)=\sigma(x)\\ \sigma_1'(x)=\sigma'(x)}} \left(\prod_{\substack{\mu\in\mathbf{Var}(\theta)\\ j=1,\ldots,k_\mu}} \omega_\mu(\sigma_1(y_{(\mu,j)}), \sigma_1'(y_{(\mu,j)}))\right) \cdot \\
&\qquad \max\left(\max_{x\in\mathsf{Var}(\theta)} d_2(\sigma_1(x), \sigma_1'(x)), \max_{\substack{\mu\in\mathbf{Var}(\theta)\\ j=1,\ldots,k_\mu}} d_2(\sigma_1(y_{(\mu,j)}), \sigma_1'(y_{(\mu,j)}))\right) \\
&\quad \text{(by } \omega_x(\sigma_1(x), \sigma_1'(x)) = 1 \text{ whenever } \sigma_1(x) = \sigma(x) \text{ and } \sigma_1'(x) = \sigma'(x), \text{ and}
\end{aligned}$$

$$
\begin{aligned}
&\omega_x(\sigma_1(x),\sigma_1'(x)) = 0 \text{ otherwise)}\\
\leq &\left(\max_{x\in \mathsf{Var}(\theta)} d_2(\sigma(x),\sigma'(x))\right) +\\
&\sum_{\substack{\sigma_1,\sigma_1'\in \mathbf{T}(\Sigma)^{\mathcal{V}_s}\\ \sigma_1(x)=\sigma(x)\\ \sigma_1'(x)=\sigma'(x)}} \left(\prod_{\substack{\mu\in \mathbf{Var}(\theta)\\ j=1,\ldots,k_\mu}} \omega_\mu(\sigma_1(y_{(\mu,j)}),\sigma_1'(y_{(\mu,j)}))\right) \cdot \max_{\substack{\mu\in \mathbf{Var}(\theta)\\ j=1,\ldots,k_\mu}} d_2(\sigma_1(y_{(\mu,j)}),\sigma_1'(y_{(\mu,j)}))\\
\leq &\left(\max_{x\in \mathsf{Var}(\theta)} d_2(\sigma(x),\sigma'(x))\right) +\\
&\sum_{\substack{\sigma_1,\sigma_1'\in \mathbf{T}(\Sigma)^{\mathcal{V}_s}\\ \sigma_1(x)=\sigma(x)\\ \sigma_1'(x)=\sigma'(x)}} \left(\prod_{\substack{\mu\in \mathbf{Var}(\theta)\\ j=1,\ldots,k_\mu}} \omega_\mu(\sigma_1(y_{(\mu,j)}),\sigma_1'(y_{(\mu,j)}))\right) \cdot \sum_{\substack{\mu\in \mathbf{Var}(\theta)\\ j=1,\ldots,k_\mu}} d_2(\sigma_1(y_{(\mu,j)}),\sigma_1'(y_{(\mu,j)}))\\
\leq &\left(\max_{x\in \mathsf{Var}(\theta)} d_2(\sigma(x),\sigma'(x))\right) +\\
&\sum_{\substack{\mu\in \mathbf{Var}(\theta)\\ j=1,\ldots,k_\mu}} \sum_{\substack{\sigma_1,\sigma_1'\in \mathbf{T}(\Sigma)^{\mathcal{V}_s}\\ \sigma_1(x)=\sigma(x)\\ \sigma_1'(x)=\sigma'(x)}} d_2(\sigma_1(y_{\mu,j}),\sigma_1'(y_{(\mu,j)})) \cdot \omega_\mu(\sigma_1(y_{(\mu,j)}),\sigma_1'(y_{(\mu,j)}))\\
= &\left(\max_{x\in \mathsf{Var}(\theta)} d_2(\sigma(x),\sigma'(x))\right) + \sum_{\substack{\mu\in \mathbf{Var}(\theta)\\ j=1,\ldots,k_\mu}} \mathbf{K}(d_2)(\sigma_1(\mu),\sigma_1'(\mu))\\
= &\max_{x\in \mathcal{V}_s} d_2(\sigma_1(x),\sigma_1'(x)) \cdot \mathrm{Var}(x,\theta) + \sum_{\mu\in \mathcal{V}_d} \mathbf{K}(d_2)(\sigma_1(\mu),\sigma_1'(\mu)) \cdot \mathrm{Var}(\mu,\theta)
\end{aligned}
$$

(since $\mathrm{Var}(x,\theta) = 1$ for all $x \in \mathsf{Var}(\theta)$ and $k_\mu = \mathrm{Var}(\mu,\theta)$ for all $\mu \in \mathsf{Var}(\theta)$).

thus confirming that ω_θ satisfies Equation 4.4.

We conclude the proof by considering the induction step $\theta = \sum_{i\in I} p_i\theta_i$. Remind that the convex combination operator is the outermost operator for distribution terms in normal form. Then by Proposition 2.32.3 and by the induction hypothesis (Equation 4.4) we get

$$
\begin{aligned}
&\mathbf{K}(d_2)(\sigma(\theta),\sigma'(\theta))\\
\leq &\sum_{i\in I} p_i\, \mathbf{K}(d_2)(\sigma(\theta_i),\sigma'(\theta_i))\\
\leq &\sum_{i\in I} p_i \left(\max_{x\in \mathcal{V}_s} d_2(\sigma(x),\sigma'(x)) \cdot \mathrm{Var}(x,\theta_i) + \sum_{\mu\in \mathcal{V}_d} \mathbf{K}(d_2)(\sigma(\mu),\sigma'(\mu)) \cdot \mathrm{Var}(\mu,\theta_i)\right)\\
= &\max_{x\in \mathcal{V}_s} d_2(\sigma(x),\sigma'(x)) \cdot \mathrm{Var}(x,\theta) + \sum_{\mu\in \mathcal{V}_d} \mathbf{K}(d_2)(\sigma(\mu),\sigma'(\mu)) \cdot \mathrm{Var}(\mu,\theta)
\end{aligned}
$$

which is precisely the proof obligation in Equation 4.4. □

Lemmas 4.8 and 4.9 provide bounds on the distance between non-extensive state and distribution terms. Now we can prove the main Theorem 4.5.

Proof of Theorem 4.5. Let d be the non-extensive congruence closure of the λ-bisimilarity metric $\mathbf{d}$ (Definition 4.7). To prove the thesis it is enough to show that d is a prefixed point of $\mathbf{B}$ on $([0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)}, \sqsubseteq)$, i.e. $\mathbf{B}(d) \sqsubseteq d$. Then, from $d \sqsubseteq \mathbf{d}$ and the fact that $\mathbf{d}$ is the least prefixed point of $\mathbf{B}$ on $([0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)}, \sqsubseteq)$ it follows that $d = \mathbf{d}$. Finally, since by construction of d all operators in Σ are non-extensive w.r.t. d, we can conclude that each operator $f \in \Sigma$ is non-extensive w.r.t. $\mathbf{d}$.

In order to show $\mathbf{B}(d) \sqsubseteq d$, it suffices to prove that d satisfies the transfer condition of bisimulation metric

$$\forall(t,a,\pi) \in \longrightarrow . \exists(t',a,\pi') \in \longrightarrow . \lambda \cdot \mathbf{K}(d)(\pi,\pi') \leq d(t,t') \tag{4.5}$$

for all $t, t' \in \mathsf{T}(\Sigma)$ with $d(t,t') < 1$. If t and t' have different outermost function symbols, then by the definition of congruence closure (Definition 4.6) we have $d(t,t') = \mathbf{d}(t,t')$. Since the λ-bisimilarity metric $\mathbf{d}$ satisfies the transfer condition of bisimulation metric we have that for each $t \xrightarrow{a} \pi$ there exists a transition $t' \xrightarrow{a} \pi'$ with $\lambda \cdot \mathbf{K}(\mathbf{d})(\pi,\pi') \leq \mathbf{d}(t,t')$. Now by $d \sqsubseteq \mathbf{d}$ and monotonicity of $\mathbf{K}$ (Proposition 2.32.1) we get

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

and Equation 4.5 is given.

Hence, it remains to show that for any given open term $t \in \mathbb{T}(\Sigma)$ and closed substitutions $\underline{\sigma}, \underline{\sigma}'$ with $\underline{\sigma}(x)$ and $\underline{\sigma}'(x)$ having different outermost function symbols for all $x \in \mathsf{Var}(t)$, the transfer condition of bisimulation metric (Equation 4.5) is satisfied for terms $\underline{\sigma}(t)$ and $\underline{\sigma}'(t)$. We will show this by structural induction over t. The base case $t = x$ is trivial and follows precisely from the earlier argumentation where $\underline{\sigma}(x)$ and $\underline{\sigma}'(x)$ had different outermost function symbols.

The induction step $t = f(t_1,\ldots,t_n)$ requires to distinct two subcases. The first subcase $d(\underline{\sigma}(t),\underline{\sigma}'(t)) = \mathbf{d}(\underline{\sigma}(t),\underline{\sigma}'(t))$ (first argument of the min operator in Definition 4.6) is trivial and follows precisely from the earlier argumentation where t and t' had different outermost function symbols. The second subcase $d(\underline{\sigma}(t),\underline{\sigma}'(t)) = \max_{i=1}^{n} d(\underline{\sigma}(t_i),\underline{\sigma}'(t_i))$ (second argument of the min operator in Definition 4.6) will be shown assuming as induction hypothesis that the transfer condition of bisimulation metric is satisfied for $\underline{\sigma}(t_i)$ and $\underline{\sigma}'(t_i)$ for all $i = 1,\ldots,n$, i.e.

$$\forall(\underline{\sigma}(t_i),a_{i,m},\pi_{i,m}) \in \longrightarrow . \exists(\underline{\sigma}'(t_i),a_{i,m},\pi'_{i,m}) \in \longrightarrow . \lambda \cdot \mathbf{K}(d)(\pi_{i,m},\pi'_{i,m}) \leq d(\underline{\sigma}(t_i),\underline{\sigma}'(t_i)) \tag{4.6}$$

if $d(\underline{\sigma}(t_i),\underline{\sigma}'(t_i)) < 1$. We use symbols $a_{i,m}$ and $\pi_{i,m}, \pi'_{i,m}$ to allow for an easy match with the rule r (Equation 4.7 below) that specifies the respective transitions.

Suppose the transition $\underline{\sigma}(t) \xrightarrow{a} \pi$ is derived from the λ-non-extensive PGSOS-rule $r \in R$ given by

$$\frac{\{x_i \xrightarrow{a_{i,m}} \mu_{i,m} \mid i \in I, m \in M_i\} \;\; \{x_i \xrightarrow{b_{i,n}} \not\rightarrow \mid i \in I, n \in N_i\}}{f(x_1,\ldots,x_n) \xrightarrow{a} \theta} \tag{4.7}$$

with the substitution σ defined by $\sigma(x_i) = \underline{\sigma}(t_i)$ for $i \in I$. Notice that $\sigma(\theta) = \pi$.

In the remainder we will first give suitable moves $\underline{\sigma}'(t_i) \xrightarrow{a_{i,m}} \pi'_{i,m}$. This allows us to define the substitution σ' as $\sigma'(x_i) = \underline{\sigma}'(t_i)$ and $\sigma'(\mu_{i,m}) = \pi'_{i,m}$. Subsequently we show

that $\sigma'(x_i) \xrightarrow{b_{i,n}}\!\!\!\!\!\!/\;$ for all $i \in I$ and $n \in N_i$. Thus, all premises of r are satisfied and the transition $\sigma'(f(x_1,\ldots,x_n)) \xrightarrow{a} \sigma'(\theta)$ can be derived from r with substitution σ'. Finally we show

$$\lambda \cdot \mathbf{K}(d)(\sigma(\theta),\sigma'(\theta)) \leq d(\sigma(f(x_1,\ldots,x_n)),\sigma'(f(x_1,\ldots,x_n))) \tag{4.8}$$

which confirms that $\sigma'(\theta)$ is the distribution π' we were looking for.

Consider the positive premises $x_i \xrightarrow{a_{i,m}} \mu_{i,m}$ of rule r. Since $d(\underline{\sigma}(t_i),\underline{\sigma'}(t_i)) < 1$ we get from the inductive hypothesis (Equation 4.6) that there is a transition $\underline{\sigma'}(t_i) \xrightarrow{a_{i,m}} \pi'_{i,m}$ with $\lambda \cdot \mathbf{K}(d)(\sigma(\mu_{i,m}),\pi'_{i,m}) \leq d(\underline{\sigma}(t_i),\underline{\sigma'}(t_i))$. Define $\sigma'(\mu_{i,m}) = \pi'_{i,m}$.

Consider the negative premises $x_i \xrightarrow{b_{i,n}}\!\!\!\!\!\!/\;$ of rule r. Since $d(\underline{\sigma}(t_i),\underline{\sigma'}(t_i)) < 1$ we get by Proposition 2.30.2 that $\underline{\sigma'}(x_i) \xrightarrow{b_{i,n}}\!\!\!\!\!\!/\;$.

To summarize, all premises of r are satisfied by σ'. It follows that the transition $\sigma'(f(x_1,\ldots,x_n)) \xrightarrow{a} \sigma'(\theta)$ can be derived. It remains to show the proof obligation Equation 4.8. From Lemma 4.9 we derive

$$\mathbf{K}(d)(\sigma(\theta),\sigma'(\theta)) \leq \max_{x\in\mathcal{V}_s}(d(\sigma(x),\sigma'(x)) \cdot \mathrm{Var}(x,\theta)) + \sum_{\mu\in\mathcal{V}_d} \mathbf{K}(d)(\sigma(\mu),\sigma'(\mu)) \cdot \mathrm{Var}(\mu,\theta) \tag{4.9}$$

which implies Equation 4.8 by

$$\begin{aligned}
&\lambda \cdot \mathbf{K}(d)(\sigma(\theta),\sigma'(\theta)) \\
\leq &\max_{x\in\mathcal{V}_s} \lambda \cdot d(\sigma(x),\sigma'(x)) \cdot \mathrm{Var}(x,\theta) + \sum_{\mu\in\mathcal{V}_d} \lambda \cdot \mathbf{K}(d)(\sigma(\mu),\sigma'(\mu)) \cdot \mathrm{Var}(\mu,\theta) \\
= &\max_{i\in I} \lambda \cdot d(\sigma(x_i),\sigma'(x_i)) \cdot \mathrm{Var}(x_i,\theta) + \sum_{\substack{i\in I \\ \mu\in \mathrm{der}(r,x_i)}} \lambda \cdot \mathbf{K}(d)(\sigma(\mu),\sigma'(\mu)) \cdot \mathrm{Var}(\mu,\theta) \\
\leq &\max_{i\in I} \lambda \cdot d(\sigma(x_i),\sigma'(x_i)) \cdot \mathrm{Var}(x_i,\theta) + \sum_{\substack{i\in I \\ \mu\in \mathrm{der}(r,x_i)}} d(\sigma(x_i),\sigma'(x_i)) \cdot \mathrm{Var}(\mu,\theta) \\
\leq &\left(\max_{i\in I} d(\sigma(x_i),\sigma'(x_i))\right) \cdot \left(\max_{i\in I} \lambda \cdot \mathrm{Var}(x_i,\theta) + \sum_{\substack{i\in I \\ \mu\in \mathrm{der}(r,x_i)}} \mathrm{Var}(\mu,\theta)\right) \\
= &\left(\max_{i\in I} d(\sigma(x_i),\sigma'(x_i))\right) \cdot \mathrm{copy}(r,\lambda) \\
\leq &\max_{i\in I} d(\sigma(x_i),\sigma'(x_i)) \\
= &d(\sigma(f(x_1,\ldots,x_n)),\sigma'(f(x_1,\ldots,x_n))).
\end{aligned}$$

with step 3 by the fact that $\sigma(x_i)$ and $\sigma'(x_i)$ satisfy the λ-bisimulation metric transfer condition, step 5 by definition of copy, step 6 by property $\mathrm{copy}(r,\lambda) \leq 1$ satisfied by the λ-non extensive rule r (Definition 4.4), and step 7 from the assumption $\max_{i=1}^{n} d(\underline{\sigma}(t_i),\underline{\sigma'}(t_i)) = d(\underline{\sigma}(t),\underline{\sigma'}(t))$ and the equalities $\underline{\sigma}(t_i) = \sigma(x_i)$, $\underline{\sigma'}(t_i) = \sigma'(x_i)$, $\underline{\sigma}(t) = \sigma(f(x_1,\ldots,x_n))$ and $\underline{\sigma'}(t) = \sigma'(f(x_1,\ldots,x_n))$.

We conclude by observing that the transition $\sigma(f(x_1,\ldots,x_n)) \xrightarrow{a} \sigma(\mathrm{trgt}(r))$ derived from the f-defining rule r can be mimicked by $\sigma'(f(x_1,\ldots,x_n)) \xrightarrow{a} \sigma'(\mathrm{trgt}(r))$ such that the metric bisimulation transfer condition

$$\begin{aligned}&\lambda\,\mathbf{K}(d)(\sigma(\mathrm{trgt}(r)),\sigma'(\mathrm{trgt}(r)))\\ \leq\,&d(\sigma(f(x_1,\ldots,x_n)),\sigma'(f(x_1,\ldots,x_n)))\\ =\,&\max_{i=1}^{n} d(\sigma(x_i),\sigma'(x_i))\end{aligned}$$

holds. Hence, operator f is non-extensive. □

4.2.3 Non-extensive process algebra operators

Theorem 4.5 allows us to determine which process algebra operators are non-extensive by inspecting their respective specification rules and verifying that the rule constraints of Definition 4.4 are satisfied.

Corollary 4.10. *The process algebra operators*

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

(specified in Figure 4.2) are non-extensive w.r.t. λ-bisimilarity metric $\mathbf{d}$ *for any* $\lambda \in (0,1]$.

Proof. By simple inspection of the specification rules in Figure 4.2 and validation of the rule constraint given in Definition 4.4. □

This result coincides with Theorem 3.5. However, while the result of Theorem 3.5 was derived by a detailed analysis of the distance between composed processes, we exploit here the structural properties of the specification rules. More general, the specification format provides a simple method to verify the compositionality properties of arbitrary (non-standard, experimental, domain-specific) process algebra or programming language operators by solely inspecting the structural properties of their respective specifications.

On the other hand, none of the operators in the Figures 4.3 and 4.4 (except the skip process constant ε) is specified by only λ-non-extensive rules, for any $\lambda \in (0,1]$. It is easy to show that all those operators are not non-extensive w.r.t. $\mathbf{d}$. As example, we consider the synchronous parallel composition operator, whose specification rule r is not λ-non extensive for any $\lambda \in (0,1]$ by $\mathrm{copy}(r,\lambda) = 2$.

Example 4.11. Consider the term $t = x \mid x'$ and let σ_1 and σ_2 be the closed substitutions defined by $\sigma_1(x) = \sigma_1(x') = a.a.0$, and $\sigma_2(x) = a.([1-\epsilon]a.0 \oplus [\epsilon]0)$, $\sigma_2(x') = a.([1-\epsilon']a.0 \oplus [\epsilon']0)$ with any fixed $\epsilon,\epsilon' \in (0,1)$. We have $\mathbf{d}(\sigma_1(x),\sigma_2(x)) = \lambda\epsilon$ and $\mathbf{d}(\sigma_1(x'),\sigma_2(x')) = \lambda\epsilon'$. Then $\mathbf{d}(\sigma_1(t),\sigma_2(t)) = \lambda(1-(1-\epsilon)(1-\epsilon')) > \max(\lambda\epsilon,\lambda\epsilon')$ for any $\lambda \in (0,1]$. Hence, the synchronous parallel composition operator $\mid$ is not non-extensive w.r.t. $\mathbf{d}$ for any $\lambda \in (0,1]$.

4.2.4 Distance between non-extensive terms

We call a term t non-extensive if all operators used in t are non-extensive. The compositionality properties of the operators in t allow us to give an upper bound on the distance between closed instances of t.

Proposition 4.12. *Let $P_1 = (\Sigma_1, A, R_1)$ be a λ-non-extensive PTSS and $P_2 = (\Sigma_2, A, R_2)$ be any PGSOS PTSS with $P_1 \sqsubseteq P_2$. Then for any term $t \in \mathbb{T}(\Sigma_1)$ we have*

$$\mathbf{d}(\sigma(t), \sigma'(t)) \leq \max_{x \in \mathcal{V}_s} \mathrm{Var}(x, t) \cdot \mathbf{d}(\sigma(x), \sigma'(x))$$

for all closed substitutions $\sigma, \sigma' : \mathcal{V}_s \to T(\Sigma_2)$.

Proof. Follows directly from Lemma 4.8. □

It is important to note that the variables in t can be instantiated by arbitrary terms in $\mathsf{T}(\Sigma_2)$ (not necessarily built of only non-extensive operators in Σ_1).

4.3 Lipschitz continuous operators

The compositionality property of non-extensiveness discussed in the former section is very strong. This allows us to give a tight bound on the distance between closed instances of non-extensive terms (Proposition 4.12). However many operators like parallel composition and recursion are not non-extensive.

We proceed now with Lipschitz continuity which captures a wide class of non-recursive and recursive process algebra and programming language operators and provides still a powerful compositional reasoning method (cf. Section 3.4). However, we will reuse all results of the former section to give an expressive rule and specification format and tight distance bounds for specifications which consist of both Lipschitz continuous and non-extensive operators.

Definition 4.13 (Lipschitz continuous operator)**.** Let $P = (\Sigma, A, R)$ be a PTSS and $L \in \mathbb{R}_{\geq 0}$ be any fixed non-negative real. An operator $f \in \Sigma$ is *L-Lipschitz continuous* w.r.t. λ-bisimilarity metric $\mathbf{d}$ if

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

for all closed terms $s_i, t_i \in \mathsf{T}(\Sigma)$. We call f *Lipschitz continuous* w.r.t. $\mathbf{d}$ if f is L-Lipschitz continuous w.r.t. $\mathbf{d}$ for some $L \in \mathbb{R}_{\geq 0}$.

Lipschitz continuity is of great practical importance since it provides a bound on the ratio of the distance between composed systems and the distance between their parts. This is the cornerstone for metric assume-guarantee like performance validation using probabilistic process algebras (Section 3.4).

1-Lipschitz continuity is also known as *non-expansiveness* which is the most widely studied compositionality property for behavioral metric semantics (e.g. [KBL01; Des+02b;

$$\frac{}{\varepsilon \xrightarrow{\surd} \delta(0)} \qquad \frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x; y \xrightarrow{a} \mu; \delta(y)} \qquad \frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{a} \nu}{x; y \xrightarrow{a} \nu}$$

$$\frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu \quad a \neq \surd}{x \mid y \xrightarrow{a} \mu \mid \nu} \qquad \frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{\surd} \nu}{x \mid y \xrightarrow{a} \delta(0)}$$

$$\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x \,|||\, y \xrightarrow{a} \mu \,|||\, \delta(y)} \qquad \frac{y \xrightarrow{a} \nu \quad a \neq \surd}{x \,|||\, y \xrightarrow{a} \delta(x) \,|||\, \nu} \qquad \frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{\surd} \nu}{x \,|||\, y \xrightarrow{\surd} \delta(0)}$$

$$\frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu \quad a \in B \setminus \{\surd\}}{x \,||_B\, y \xrightarrow{a} \mu \,||_B\, \nu} \qquad \frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{\surd} \nu}{x \,||_B\, y \xrightarrow{\surd} \delta(0)}$$

$$\frac{x \xrightarrow{a} \mu \quad a \notin B \cup \{\surd\}}{x \,||_B\, y \xrightarrow{a} \mu \,||_B\, \delta(y)} \qquad \frac{y \xrightarrow{a} \nu \quad a \notin B \cup \{\surd\}}{x \,||_B\, y \xrightarrow{a} \delta(x) \,||_B\, \nu}$$

$$\frac{x \xrightarrow{a} \mu \quad y \not\xrightarrow{a} \quad a \neq \surd}{x \,|||_p\, y \xrightarrow{a} \mu \,|||_p\, \delta(y)} \qquad \frac{x \not\xrightarrow{a} \quad y \xrightarrow{a} \nu \quad a \neq \surd}{x \,|||_p\, y \xrightarrow{a} \delta(x) \,|||_p\, \nu}$$

$$\frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu \quad a \neq \surd}{x \,|||_p\, y \xrightarrow{a} \mu \,|||_p\, \delta(y) \oplus_p \delta(x) \,|||_p\, \nu} \qquad \frac{x \xrightarrow{\surd} \mu \quad y \xrightarrow{\surd} \nu}{x \,|||_p\, y \xrightarrow{\surd} \delta(0)}$$

Figure 4.3: Non-expansive process algebra operators

Des+04; Den+05; SDC07; Tin08; Tin10; TDZ11; GT13; Cha+14]). By definition an operator f is 1-non-extensive iff f is non-expansive. Moreover, if f is q-non-extensive for any $q \geq 1$, then f is also non-expansive. We consider the uniform notion of continuity because we aim at universal compositionality guarantees.

4.3.1 Analysis of Lipschitz continuous operators

We discuss first a few examples that analyze which rule patterns specify and which do not specify Lipschitz continuous operators. We will show the interplay between (i) the Lipschitz factors of the operators in the rule target, (ii) the probabilistic choices in the rule target, and (iii) the discount factor of the bisimilarity metric. We start with analyzing unary operators specified by rules with only non-expansive operators in the target.

Example 4.14. Consider the rules

$$\frac{x \xrightarrow{a} \mu}{f(x) \xrightarrow{a} \theta} \qquad \frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu}{x \mid y \xrightarrow{a} \mu \mid \nu}$$

with $\theta \in \mathbb{DT}(\Sigma)$ any open distribution term. For the discussion in this example we assume that the synchronous parallel composition operator $\mid$ (specified in Figure 4.3) is non-expansive (formally shown below in Corollary 4.23). Remember from Example 4.11 that $\mid$ is not non-extensive. In the following we analyze for various distribution terms θ for which $L \in \mathbb{R}_{\geq 0}$ the specified operator f is L-Lipschitz continuous. We will use the terms $s = a.a.a.0$ and $t = a.([1-\epsilon]a.a.0 \oplus [\epsilon]0)$ with ϵ some fixed value in $(0,1)$ as arguments for operator f. Then $\mathbf{d}(s,t) = \lambda\epsilon$.

Consider $\theta = \delta(x) \mid \delta(x)$. The source process x appears twice in the rule target in the context of the synchronous parallel composition operator. The transitions $f(s) \xrightarrow{a} \pi_s$ with $\pi_s = \delta(s) \mid \delta(s)$, and $f(t) \xrightarrow{a} \pi_t$ with $\pi_t = \delta(t) \mid \delta(t)$, are derivable. We have $\mathbf{K}(\mathbf{d})(\pi_s, \pi_t) = \mathbf{d}(s \mid s, t \mid t) = \lambda(1-(1-\epsilon)^2)$. Hence $\mathbf{d}(f(s), f(t)) = \lambda\,\mathbf{K}(\mathbf{d})(\pi_s, \pi_t) = \lambda^2(1-(1-\epsilon)^2) \leq \lambda^2 \cdot 2\epsilon = 2\lambda\mathbf{d}(s,t)$. Therefore, for these specific arguments s and t the L-Lipschitz continuity condition is satisfied if $L \geq 2\lambda$. Theorem 4.19 below will confirm that this specification of operator f is 2λ-Lipschitz continuous. Hence, f is non-expansive if $\lambda \leq 0.5$.

Consider $\theta = (\delta(x) \mid \delta(x)) \oplus_r \delta(0)$ for some $r \in (0,1)$. Now, the two instances of the source process x are realized only by probability r. We get $\mathbf{d}(f(s), f(t)) = r\lambda^2(1-(1-\epsilon)^2) \leq 2r\lambda\mathbf{d}(s,t)$. Theorem 4.19 below will confirm that this specification of operator f is $2r\lambda$-Lipschitz continuous. Hence, f is non-expansive if $r \cdot \lambda \leq 0.5$.

Consider $\theta = \mu \mid \mu$. The x-derivative μ appears twice in the rule target θ in the context of the synchronous parallel composition operator. The transitions $f(s) \xrightarrow{a} \pi_s$, with $\pi_s = \delta(a.a.0 \mid a.a.0)$, and $f(t) \xrightarrow{a} \pi_t$, with $\pi_t = (1-\epsilon)^2\delta(a.a.0 \mid a.a.0) + \epsilon(1-\epsilon)\delta(a.a.0 \mid 0) + \epsilon(1-\epsilon)\delta(0 \mid a.a.0) + \epsilon^2\delta(0 \mid 0)$, are derivable. Now, $\mathbf{K}(\mathbf{d})(\pi_s, \pi_t) = 1-(1-\epsilon)^2$. Thus, $\mathbf{d}(f(s), f(t)) = \lambda\,\mathbf{K}(\mathbf{d})(\pi_s, \pi_t) = \lambda(1-(1-\epsilon)^2) \leq \lambda \cdot 2\epsilon = 2\mathbf{d}(s,t)$. Therefore, for these specific arguments s and t the L-Lipschitz continuity condition is satisfied if $L \geq 2$. Theorem 4.19 below will confirm that this specification of operator f is 2-Lipschitz continuous. Moreover, for no $\lambda \in (0,1]$ this specification of operator f is non-expansive w.r.t. $\mathbf{d}$. Note that the Lipschitz factor is independent from the bisimulation discount λ since the x-derivative μ is copied and not as before the source process x itself. Probabilistic choice applies as before s.t. the operator f with rule target $\theta = (\mu \mid \mu) \oplus_r \delta(0)$ is $2r$-Lipschitz continuous. Similarly, the combination of evolution and delay of the argument process leads for operator f with rule target $\theta = (\mu \mid \delta(x)) \oplus_r \delta(0)$ to a Lipschitz factor of $(1+\lambda)r$.

In essence, Example 4.14 shows that a unary operator is L-Lipschitz continuous if at most L copies of the argument process are spawned in the transitions derived from the rules specifying that operator. Technically, if the operator is specified by rules with only non-expansive operators in the target, then the sum of the number of source variables, weighted by λ since those instances are delayed, and the number of derivatives, both weighted by the probability of their realization, may not exceed the Lipschitz factor L.

We proceed with analyzing unary operators that are specified by rules with arbitrary Lipschitz operators in the target.

Example 4.15. Consider the rules

$$\frac{x \xrightarrow{a} \mu}{f(x) \xrightarrow{a} \theta} \qquad \frac{x \xrightarrow{a} \mu}{g(x) \xrightarrow{a} \mu \mid \mu} \qquad \frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu}{x \mid y \xrightarrow{a} \mu \mid \nu}$$

with $\theta \in \mathbb{DT}(\Sigma)$ any open distribution term. For the discussion in this example we assume that, as argued in Example 4.14, operator g is 2-Lipschitz continuous and $|$ is non-expansive. We will use the terms $s = a.a.a.0$ and $t = a.a.([1-\epsilon]a.0 \oplus [\epsilon]0)$ with ϵ some fixed value in $(0,1)$ as arguments for operator f. Then $\mathbf{d}(s,t) = \lambda^2\epsilon$.

Consider $\theta = \delta(g(x))$. In the rule specifying f the source process x appears in the context of the 2-Lipschitz operator g. The transitions $f(s) \xrightarrow{a} \delta(g(s))$, and $f(t) \xrightarrow{a} \delta(g(t))$ are derivable. Since g is 2-Lipschitz continuous, we infer $\mathbf{d}(f(s), f(t)) \leq \lambda \cdot \mathbf{K}(\mathbf{d})(\delta(g(s)), \delta(g(t))) = \lambda \cdot \mathbf{d}(g(s), g(t)) \leq 2\lambda\mathbf{d}(s,t)$. Theorem 4.19 below will confirm that this specification of operator f is 2λ-Lipschitz continuous.

Consider $\theta = g(\mu)$. In the rule specifying g, the x-derivative μ appears in the context of the 2-Lipschitz continuous operator g. The transitions $f(s) \xrightarrow{a} \pi_s$, with $\pi_s = \delta(g(a.a.0))$, and $f(t) \xrightarrow{a} \pi_t$, with $\pi_t = \delta(g(a.([1-\epsilon]a.0 \oplus [\epsilon]0)))$ are derivable. Now, $\mathbf{K}(\mathbf{d})(\pi_s, \pi_t) = \lambda(1-(1-\epsilon)^2) \leq 2\lambda\epsilon$. Thus, $\mathbf{d}(f(s), f(t)) = \lambda\,\mathbf{K}(\mathbf{d})(\pi_s, \pi_t) \leq 2\lambda^2\epsilon = 2\mathbf{d}(s,t)$. Theorem 4.19 below will confirm that this specification of operator f is 2-Lipschitz continuous.

Probabilistic choice applies as in Example 4.14 s.t. the operator f with $\theta = g(\delta(x) \oplus_r \delta(0))$ is $2\lambda r$-Lipschitz continuous, and that f with $\theta = g(\mu \oplus_r \delta(0))$ is $2r$-Lipschitz continuous.

In essence, Example 4.15 confirms that a unary operator f is L-Lipschitz continuous if at most L copies of the argument process t are spawned along the evolution of the combined process $f(t)$. Technically, this means that in the target of a rule specifying a L-Lipschitz continuous operator the sum of the number of source variables, weighted by λ, and the number of derivatives, both weighted by the probability of their realization, and additionally weighted by the Lipschitz factors of respective operators, may not exceed the Lipschitz factor L.

We proceed with analyzing binary operators.

Example 4.16. Consider the rules

$$\frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu}{f(x,y) \xrightarrow{a} \theta} \qquad \frac{x \xrightarrow{a} \mu \quad y \xrightarrow{a} \nu}{x \mid y \xrightarrow{a} \mu \mid \nu}$$

with $\theta \in \mathbb{DT}(\Sigma)$ any open distribution term. Recall that the synchronous parallel composition operator $|$ is non-expansive. We consider again various distribution terms θ and analyze for which $L \in \mathbb{R}_{\geq 0}$ the specified operator f is L-Lipschitz continuous. We will use the pairs of terms (s_1, s_2) and (t_1, t_2) as arguments for operator f, with $s_1 = a.a.a.0$, $s_2 = a.a.a.0$, $t_1 = a.([1-\epsilon_1]a.a.0 \oplus [\epsilon_1]0)$ and $t_2 = a.([1-\epsilon_2]a.a.0 \oplus [\epsilon_2]0)$, with ϵ_1, ϵ_2 any fixed value in $(0,1)$. Then $\mathbf{d}(s_1, t_1) = \lambda\epsilon_1$ and $\mathbf{d}(s_2, t_2) = \lambda\epsilon_2$.

Consider $\theta = (\delta(x) \mid \delta(y)) \mid (\delta(x) \mid \delta(y))$. The source processes x and y occur each twice in the rule target in the context of the synchronous parallel composition operator. We get $\mathbf{d}(f(s_1,s_2), f(t_1,t_2)) \leq \lambda\,\mathbf{K}(\mathbf{d})(\delta((s_1 \mid s_2) \mid (s_1 \mid s_2)), \delta((t_1 \mid t_2) \mid (t_1 \mid t_2))) = \lambda\mathbf{d}((s_1 \mid s_2) \mid (s_1 \mid s_2), (t_1 \mid t_2) \mid (t_1 \mid t_2)) \leq 2\lambda\mathbf{d}(s_1 \mid s_2, t_1 \mid t_2) \leq 2\lambda(\mathbf{d}(s_1,t_1) + \mathbf{d}(s_2,t_2))$. For these specific pairs of arguments (s_1, s_2) and (t_1, t_2) the L-Lipschitz continuity condition is satisfied if $L \geq 2\lambda$. Theorem 4.19 below will confirm that this specification of operator f is 2λ-Lipschitz continuous.

Consider $\theta = (\delta(x) \mid \delta(y)) \mid (\delta(x) \mid \delta(y)) \oplus_r \delta(0)$ for some fixed $r \in (0,1)$. The probabilistic choice applies exactly as in Example 4.14. Since $\mathbf{d}(f(s_1,s_2),f(t_1,t_2)) = 2r\lambda(\mathbf{d}(s_1,t_1)+\mathbf{d}(s_2,t_2))$, the L-Lipschitz continuity condition is for these specific pairs of arguments (s_1,s_2) and (t_1,t_2) satisfied if $L \geq 2r\lambda$. Theorem 4.19 below will confirm that this specification of operator f is $2r\lambda$-Lipschitz continuous.

Consider $\theta = (\mu \mid \nu) \mid (\mu \mid \nu)$. Both the x-derivative μ and the y-derivative ν appear each twice in the context of the synchronous parallel composition operator. We get $\mathbf{d}(f(s_1,s_2),f(t_1,t_2)) = \lambda(1-(1-\epsilon_1)^2(1-\epsilon_2)^2) = \lambda(1-(1-\mathbf{d}(s_1,t_1)/\lambda)^2(1-\mathbf{d}(s_2,t_2)/\lambda)^2) \leq 2(\mathbf{d}(s_1,t_1)+\mathbf{d}(s_2,t_2))$. For these specific pairs of arguments (s_1,s_2) and (t_1,t_2) the L-Lipschitz continuity condition is satisfied if $L \geq 2$. Theorem 4.19 below will confirm that this specification of operator f is 2-Lipschitz continuous. Note again that the Lipschitz factor is independent from the bisimulation discount λ since the derivatives μ, ν are copied and not as before the source process x, y themselves. Probabilistic choice applies as before s.t. the operator f with rule target $\theta = ((\mu \mid \nu) \mid (\mu \mid \nu)) \oplus_r \delta(0)$ is $2r$-Lipschitz continuous. The combination of evolution and delay of the argument processes leads for operator f with rule target $\theta = ((\mu \mid \delta(y)) \mid (\mu \mid \delta(y))) \oplus_r \delta(0)$ to a Lipschitz factor of $\max(2, 2\lambda) \cdot r = 2r$.

In essence, Example 4.16 shows that an n-ary operator f is L-Lipschitz continuous if at most L copies of each of the processes $t_1,\ldots,t_n$ are spawned along the evolution of the composed process $f(t_1,\ldots,t_n)$. Technically, this boils down to verifying for all rules specifying the operator f, that for each source variable the sum of the number of its occurrences weighted by the discount factor and the number of occurrences of its derivatives, both weighted by the probability of their realization, and additionally weighted by the Lipschitz factors of the applied operators, may not exceed L.

4.3.2 Specification of Lipschitz continuous operators

Given any state or distribution term, the mapping Var defined in Equation 4.1 (Section 4.2) gives for each variable the number of its occurrences in that term, weighted by the probability of their realization. Now we refine this mapping by weighting the number of occurrences of some variable additionally by the Lipschitz factors of those operators that are applied on top of that variable.

Notation 4.17. Let $P_1 = (\Sigma_1, A, R_1)$ be a λ-non-extensive PTSS, and $P_2 = (\Sigma_2, A, R_2)$ with $P_1 \sqsubseteq P_2$ be any PGSOS PTSS. Let $L : \Sigma \to \mathbb{R}^{\infty}_{\geq 0}$ with $\Sigma = \Sigma_2 \setminus \Sigma_1$ be a mapping that assigns to each operator its respective Lipschitz factor, or assigns ∞ if the operator is not Lipschitz continuous. We denote by $\mathscr{L}_\Sigma$ the set of all mappings $\Sigma \to \mathbb{R}^{\infty}_{\geq 0}$.

$$\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x^{n+1} \xrightarrow{a} \mu; \delta(x^n)} \qquad \frac{x \xrightarrow{\surd} \mu}{x^{n+1} \xrightarrow{\surd} \mu} \qquad \frac{}{x^0 \xrightarrow{\surd} \delta(0)} \qquad \frac{x \xrightarrow{\surd} \mu \quad x \xrightarrow{a} \nu \quad a \neq \surd \quad n > m}{x^n \xrightarrow{a} \nu; \delta(x^m)}$$

$$\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x^\omega \xrightarrow{a} \mu; \delta(x^\omega)} \qquad \frac{x \xrightarrow{a} \mu \quad a \neq \surd}{x^* y \xrightarrow{a} \mu; \delta(x^* y)} \qquad \frac{y \xrightarrow{a} \nu}{x^* y \xrightarrow{a} \nu}$$

$$\frac{x \xrightarrow{a} \mu \;\; y \xrightarrow{a} \nu \;\; a \neq \surd}{x^{*_p} y \xrightarrow{a} \nu \oplus_p \mu; \delta(x^{*_p} y)} \qquad \frac{x \xrightarrow{a} \mu \;\; y \not\xrightarrow{a} \;\; a \neq \surd}{x^{*_p} y \xrightarrow{a} \mu; \delta(x^{*_p} y)} \qquad \frac{x \not\xrightarrow{a} \;\; y \xrightarrow{a} \nu \;\; a \neq \surd}{x^{*_p} y \xrightarrow{a} \nu} \qquad \frac{y \xrightarrow{\surd} \nu}{x^{*_p} y \xrightarrow{\surd} \nu}$$

$$\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{!^{n+1} x \xrightarrow{a} \mu \,|||\, \delta(!^n x)} \qquad \frac{x \xrightarrow{\surd} \mu}{!^{n+1} x \xrightarrow{\surd} \mu} \qquad \frac{}{!^0 x \xrightarrow{\surd} \delta(0)}$$

$$\frac{x \xrightarrow{a} \mu \quad a \neq \surd}{!x \xrightarrow{a} \mu \,|||\, \delta(!x)} \qquad \frac{x \xrightarrow{a} \mu \quad a \neq \surd}{!_p x \xrightarrow{a} \mu \oplus_p (\mu \,|||\, \delta(!_p x))}$$

Figure 4.4: Lipschitz continuous process algebra operators

The mapping $\mathrm{Var}\colon (\mathscr{V} \times (\mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma))) \to \mathbb{R}_{\geq 0}$ is now defined as follows:

$$\mathrm{Var}(\zeta, t) = \begin{cases} 1 & \text{if } t = \zeta \\ \max\limits_{i=1}^{n} \mathrm{Var}(\zeta, t_i) & \text{if } \left[\begin{array}{l} t = f(t_1, \ldots, t_n) \wedge \\ f \in \Sigma_1 \end{array}\right. \\ L(f) \sum\limits_{i=1}^{n} \mathrm{Var}(\zeta, t_i) & \text{if } \left[\begin{array}{l} t = f(t_1, \ldots, t_n) \wedge \\ f \in \Sigma_2 \setminus \Sigma_1 \end{array}\right. \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{Var}(\zeta, \theta) = \begin{cases} 1 & \text{if } \theta = \zeta \\ \mathrm{Var}(\zeta, t) & \text{if } \theta = \delta(t) \\ \sum\limits_{i \in I} p_i \cdot \mathrm{Var}(\zeta, \theta_i) & \text{if } \theta = \sum\limits_{i \in I} p_i \theta_i \\ \max\limits_{i=1}^{n} \mathrm{Var}(\zeta, \theta_i) & \text{if } \left[\begin{array}{l} \theta = f(\theta_1, \ldots, \theta_n) \wedge \\ \zeta \in \mathscr{V}_s \wedge f \in \Sigma_1 \end{array}\right. \\ \sum\limits_{i=1}^{n} \mathrm{Var}(\zeta, \theta_i) & \text{if } \left[\begin{array}{l} \theta = f(\theta_1, \ldots, \theta_n) \wedge \\ \zeta \in \mathscr{V}_d \wedge f \in \Sigma_1 \end{array}\right. \\ L(f) \sum\limits_{i=1}^{n} \mathrm{Var}(\zeta, \theta_i) & \text{if } \left[\begin{array}{l} \theta = f(\theta_1, \ldots, \theta_n) \wedge \\ \zeta \in \mathscr{V}_s \wedge f \in \Sigma_2 \setminus \Sigma_1 \end{array}\right. \\ \overline{L(f)} \sum\limits_{i=1}^{n} \mathrm{Var}(\zeta, \theta_i) & \text{if } \left[\begin{array}{l} \theta = f(\theta_1, \ldots, \theta_n) \wedge \\ \zeta \in \mathscr{V}_d \wedge f \in \Sigma_2 \setminus \Sigma_1 \end{array}\right. \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

with $\overline{L(f)} = \max(L(f), 1)$. Note that Var applied to non-extensive terms, i.e. only operators of Σ_1, coincides with the former definition in Equation 4.1. Case 5 and case 7 with $L(f) < 1$ of the $\mathrm{Var}(\zeta, \theta)$ definition capture the application of operators f to distribution terms where f has a modulus of continuity on state terms that is strictly below 1-Lipschitz continuity. As shown in Example 5.37 (Section 5.3.3) this yields a modulus of continuity of 1-Lipschitz continuity on the respective distribution terms.

Using the refined definition of Var we define now the mapping $\mathsf{copy} : R \times (0,1] \times \mathcal{V}_s \to \mathbb{R}^{\infty}_{\geq 0}$ as:

$$\mathsf{copy}(r, \lambda, x_i) = \lambda \cdot \mathrm{Var}(x_i, \mathrm{trgt}(r)) + \sum_{\mu \in \mathrm{der}(r, x_i)} \mathrm{Var}(\mu, \mathrm{trgt}(r)). \tag{4.11}$$

Given some rule r, the expression $\mathsf{copy}(r, \lambda, x_i)$ describes how many copies of the source x_i and its derivatives $\mu \in \mathrm{der}(r, x_i)$ emerge in the transition specified by r. An operator is L-Lipschitz continuous if at most L instances of each source process or its derivatives (up to probabilistic weighting and discounting) evolve.

Definition 4.18 (Lipschitz continuity format)**.** Let P_1, P_2, L as in notation 4.17. A rule $r \in R_2$ that specifies some operator f is a (L, λ)-*Lipschitz rule* if

$$\mathsf{copy}(r, \lambda, x_i) \leq L(f)$$

for all source variables $x_i \in \{x_1, \ldots, x_n\}$ of r. We call P_2 a (L, λ)-*Lipschitz PTSS* if all rules $r \in R_2$ are (L, λ)-Lipschitz rules.

To understand the rule format, observe that $L(f) = \mathrm{Var}(x_i, f(x_1, \ldots, x_n))$ for any $x_i \in \{x_1, \ldots, x_n\}$. Then, the condition is $\mathsf{copy}(r, \lambda, x_i) \leq \mathrm{Var}(x_i, f(x_1, \ldots, x_n))$. This mimics precisely the transfer condition of the bisimulation metric in the sense that the distance between two instances of $f(x_1, \ldots, x_n)$ is at least the distance between the accessible distributions which are instances of $\mathrm{trgt}(r)$.

Lipschitz PTSS specify Lipschitz continuous operators.

Theorem 4.19. *Let P_1, P_2, L as in notation 4.17 with P_2 a (L, λ)-Lipschitz PTSS. Then all operators $f \in \Sigma$ with $L(f) < \infty$ are $L(f)$-Lipschitz-continuous w.r.t. λ-bisimilarity metric* **d**.

We prove Theorem 4.19 by following the same line of argumentation as in the proof of Theorem 4.5.

Definition 4.20 (Lipschitz congruence closure)**.** Let P_1, P_2, L as in notation 4.17 and $d : \mathsf{T}(\Sigma_2) \times \mathsf{T}(\Sigma_2) \to [0,1]$ be any function. We call $\mathsf{cl}_m(d)$ with $(m)_{f \in \Sigma_2}$ defined by

$$m_f(\epsilon_1, \ldots, \epsilon_n) = \begin{cases} \max_{i=1}^{n} \epsilon_i & \text{if } f \in \Sigma_1 \\ L(f) \sum_{i=1}^{n} \epsilon_i & \text{if } f \in \Sigma_2 \setminus \Sigma_1 \end{cases}$$

the *Lipschitz congruence closure* of d w.r.t. Σ_1, Σ_2, L.

Again, we define first an upper bound on the distance between state and distribution terms composed of either non-extensive or Lipschitz continuous operators.

Lemma 4.21. *Let P_1, P_2, L as in notation 4.17, $d_1 \colon T(\Sigma_2) \times T(\Sigma_2) \to [0,1]$ be any function, and d_2 the Lipschitz congruences closure of d_1 w.r.t. Σ_1, Σ_2, L. Then, for any term $t \in \mathbb{T}(\Sigma_2)$ we have*

$$d_2(\sigma(t), \sigma'(t)) \leq \sum_{x \in \mathcal{V}_s} d_2(\sigma(x), \sigma'(x)) \cdot \mathrm{Var}(x, t)$$

for all closed substitutions $\sigma, \sigma' \colon \mathcal{V}_s \to T(\Sigma)$.

Proof. By structural induction over t. The base case $t = x \in \mathcal{V}_s$ follows immediately by $\mathrm{Var}(x, x) = 1$. Consider the induction step $t = f(t_1, \ldots, t_n)$. If $f \in \Sigma_1$, then we have

$$\begin{aligned}
& d_2(\sigma(t), \sigma'(t)) \\
\leq & \max_{i=1,\ldots,n} d_2(\sigma(t_i), \sigma'(t_i)) && \text{(definition of } d_2\text{)} \\
\leq & \max_{i=1,\ldots,n} \sum_{x \in \mathcal{V}_s} d_2(\sigma(x), \sigma'(x)) \cdot \mathrm{Var}(x, t_i) && \text{(inductive hypothesis)} \\
\leq & \sum_{x \in \mathcal{V}_s} d_2(\sigma(x), \sigma'(x)) \cdot \left(\max_{i=1,\ldots,n} \mathrm{Var}(x, t_i) \right) \\
= & \sum_{x \in \mathcal{V}_s} d_2(\sigma(x), \sigma'(x)) \cdot \mathrm{Var}(x, t) && \text{(definition of Var).}
\end{aligned}$$

Otherwise, if $f \in \Sigma_2 \setminus \Sigma_1$, then we have

$$\begin{aligned}
& d_2(\sigma(t), \sigma'(t)) \\
\leq & L(f) \sum_{i=1,\ldots,n} d_2(\sigma(t_i), \sigma'(t_i)) && \text{(definition of } d_2\text{)} \\
\leq & L(f) \sum_{i=1,\ldots,n} \sum_{x \in \mathcal{V}_s} d_2(\sigma(x), \sigma'(x)) \cdot \mathrm{Var}(x, t_i) && \text{(inductive hypothesis)} \\
= & \sum_{x \in \mathcal{V}_s} d_2(\sigma(x), \sigma'(x)) \left(L(f) \sum_{i=1,\ldots,n} \mathrm{Var}(x, t_i) \right) \\
= & \sum_{x \in \mathcal{V}_s} d_2(\sigma(x), \sigma'(x)) \cdot \mathrm{Var}(x, t) && \text{(definition of Var).}
\end{aligned}$$

□

Lemma 4.22. *Let P_1, P_2, L as in notation 4.17, $d_1 \colon T(\Sigma_2) \times T(\Sigma_2) \to [0,1]$ be any function, and d_2 the Lipschitz congruence closure of d_1 w.r.t. Σ_1, Σ_2, L. Then, for any distribution term $\theta \in \mathbb{DT}(\Sigma_2)$ we have*

$$\mathbf{K}(d_2)(\sigma(\theta), \sigma'(\theta)) \leq \sum_{x \in \mathcal{V}_s} d_2(\sigma(x), \sigma'(x)) \cdot \mathrm{Var}(x, \theta) + \sum_{\mu \in \mathcal{V}_d} \mathbf{K}(d_2)(\sigma(\mu), \sigma'(\mu)) \cdot \mathrm{Var}(\mu, \theta)$$

for all closed substitutions $\sigma, \sigma' \colon \mathcal{V} \to T(\Sigma) \cup DT(\Sigma)$.

Proof. The proof of this lemma follows the same line of argumentation as the proof of Lemma 4.9. Without loss of generality, assume that θ is in normal form (Definition 2.8 and Proposition 2.9). We proceed by induction over θ. The base case $\theta = \mu$

follows immediately by $\mathrm{Var}(\mu,\mu) = 1$. For the base case $\theta = \delta(x)$ the thesis follows by $\mathbf{K}(d_2)(\sigma(\delta(x)),\sigma'(\delta(x))) \leq d_2(\sigma(x),\sigma'(x))$ (Proposition 2.32.2), $d_2(\sigma(x),\sigma'(x)) \leq \sum_{x\in\mathscr{V}_s} d_2(\sigma(x),\sigma'(x))\cdot\mathrm{Var}(x,x)$ (Lemma 4.21) and $\mathrm{Var}(x,x) = \mathrm{Var}(x,\delta(x))$.

Consider the induction step $\theta = f(\theta_1,\ldots,\theta_n)$. If $f \in \Sigma_2\backslash\Sigma_1$, then by definition of d_2 as Lipschitz congruence closure of d_1 w.r.t. Σ_1,Σ_2,L we get $d_2(f(s_1,\ldots,s_n),f(t_1,\ldots,t_n)) \leq L(f)\sum_{i=1}^n d_2(s_i,t_i)$ for all $s_i,t_i \in \mathsf{T}(\Sigma)$. Hence we get

$$
\begin{aligned}
&\mathbf{K}(d_2)(\sigma(f(\theta_1,\ldots,\theta_n)),\sigma'(f(\theta_1,\ldots,\theta_n)))\\
\leq& L(f)\sum_{i=1}^n \mathbf{K}(d_2)(\sigma(\theta_i),\sigma'(\theta_i))\\
\leq& L(f)\sum_{i=1}^n\left(\sum_{x\in\mathscr{V}_s} d_2(\sigma(x),\sigma'(x))\cdot\mathrm{Var}(x,\theta_i) + \sum_{\mu\in\mathscr{V}_d}\mathbf{K}(d_2)(\sigma(\mu),\sigma'(\mu))\cdot\mathrm{Var}(\mu,\theta_i)\right)\\
=& \sum_{x\in\mathscr{V}_s} d_2(\sigma(x),\sigma'(x))L(f)\sum_{i=1}^n\mathrm{Var}(x,\theta_i) + \sum_{\mu\in\mathscr{V}_d}\mathbf{K}(d_2)(\sigma(\mu),\sigma'(\mu))L(f)\sum_{i=1}^n\mathrm{Var}(\mu,\theta_i)\\
\leq& \sum_{x\in\mathscr{V}_s} d_2(\sigma(x),\sigma'(x))L(f)\sum_{i=1}^n\mathrm{Var}(x,\theta_i) + \sum_{\mu\in\mathscr{V}_d}\mathbf{K}(d_2)(\sigma(\mu),\sigma'(\mu))\overline{L(f)}\sum_{i=1}^n\mathrm{Var}(\mu,\theta_i)\\
=& \sum_{x\in\mathscr{V}_s} d_2(\sigma(x),\sigma'(x))\cdot\mathrm{Var}(x,\theta) + \sum_{\mu\in\mathscr{V}_d}\mathbf{K}(d_2)(\sigma(\mu),\sigma'(\mu))\cdot\mathrm{Var}(\mu,\theta).
\end{aligned}
$$

with step 1 by Corollary 2.34, step 2 by the induction hypothesis on θ_i and the last step by the definition of Var.

If $f \in \Sigma_1$, then the reasoning follows precisely the argumentation and proof steps of Lemma 4.9. In summary, first we construct an appropriate matching $\omega_\theta \in \Omega(\sigma(\theta),\sigma'(\theta))$ for $\sigma(\theta)$ and $\sigma'(\theta)$, then a state term t_θ such that all terms in the support of $\sigma(\theta)$ and $\sigma'(\theta)$ are instances of t_θ, and finally show that proof obligation in the two steps $\mathbf{K}(d_2)(\sigma(\theta),\sigma'(\theta)) \leq \sum_{t,t'\in\mathsf{T}(\Sigma)}\omega_\theta(t,t')\cdot d_2(t,t')$ and $\sum_{t,t'\in\mathsf{T}(\Sigma)}\omega_\theta(t,t')\cdot d_2(t,t') \leq \sum_{x\in\mathscr{V}_s} d_2(\sigma(x),\sigma'(x))\cdot\mathrm{Var}(x,\theta) + \sum_{\mu\in\mathscr{V}_d}\mathbf{K}(d_2)(\sigma(\mu),\sigma'(\mu))\cdot\mathrm{Var}(\mu,\theta)$.

The final case $\theta = \sum_{i\in I} p_i\theta_i$ follows precisely the proof steps of Lemma 4.9. □

Now we can show the main Theorem 4.19.

Proof of Theorem 4.19. The reasoning will follow the same line of argumentation as the proof of Theorem 4.5. Assume notation 4.17 and let d be the Lipschitz congruence closure of the λ-bisimilarity metric $\mathbf{d}$ w.r.t. Σ_1,Σ_2,L (Definition 4.20). As before, we show that d is a prefixed point of $\mathbf{B}$ on $([0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)},\sqsubseteq)$ by proving that d satisfies the transfer condition of bisimulation metric

$$\forall(t,a,\pi)\in\longrightarrow.\ \exists(t',a,\pi')\in\longrightarrow.\ \lambda\,\mathbf{K}(d)(\pi,\pi') \leq d(t,t') \tag{4.12}$$

for all $t,t' \in \mathsf{T}(\Sigma_2)$ with $d(t,t') < 1$. The thesis then follows by the same argument in the proof of Theorem 4.5.

If t and t' have different outermost function symbols, then Equation 4.12 follows as in the case of Theorem 4.5.

Hence, it remains to show that for any given open term $t \in \mathbb{T}(\Sigma)$ and closed substitutions $\underline{\sigma}, \underline{\sigma}'$ with $\underline{\sigma}(x)$ and $\underline{\sigma}'(x)$ having different outermost function symbols for all $x \in \mathsf{Var}(t)$, the transfer condition of bisimulation metric (Equation 4.12) is satisfied for terms $\underline{\sigma}(t)$ and $\underline{\sigma}'(t)$. We will show this by structural induction over t. The base case $t = x$ is trivial and follows precisely from the earlier argumentation where $\underline{\sigma}(x)$ and $\underline{\sigma}'(x)$ had different outermost function symbols.

The induction step $t = f(t_1, \ldots, t_n)$ requires three subcases. The first subcase given by $d(\underline{\sigma}(t), \underline{\sigma}'(t)) = \mathbf{d}(\underline{\sigma}(t), \underline{\sigma}'(t))$ (first argument of the min operator in Definition 4.6) is trivial and follows precisely from the earlier argumentation where t and t' had different outermost function symbols. The second subcase $d(\underline{\sigma}(t), \underline{\sigma}'(t)) = \max_{i=1}^{n} d(\underline{\sigma}(t_i), \underline{\sigma}'(t_i))$ (second argument of the min operator in Definition 4.6 and, then, case $f \in \Sigma_1$ in Definition 4.20) follows as in Theorem 4.5.

The third remaining subcase is $d(\underline{\sigma}(t), \underline{\sigma}'(t)) = L(f) \sum_{i=1}^{n} d(\underline{\sigma}(t_i), \underline{\sigma}'(t_i))$ (second argument of the min operator in Definition 4.6 and, then, case $f \in \Sigma_2 \setminus \Sigma_1$ in Definition 4.20). Assume $d(\underline{\sigma}(t), \underline{\sigma}'(t)) < 1$. Let σ be any closed substitution with $\sigma(x_i) = \underline{\sigma}(t_i)$ and r be any (L, λ)-Lipschitz rule defining operator f with $\theta = \mathrm{trgt}(r)$ such that the transition $\sigma(f(x_1, \ldots, x_n)) \xrightarrow{a} \sigma(\theta)$ is derivable from r by σ. Like in the proof of Theorem 4.5 we construct an appropriate closed substitution σ' with $\sigma'(x_i) = \underline{\sigma}'(t_i)$ such that a transition $\sigma'(f(x_1, \ldots, x_n)) \xrightarrow{a} \sigma'(\theta)$ can be derived from r by σ' for which the metric transfer condition

$$\lambda \cdot \mathbf{K}(d)(\sigma(\theta), \sigma'(\theta)) \leq d(\sigma(f(x_1, \ldots, x_n)), \sigma'(f(x_1, \ldots, x_n))) \tag{4.13}$$

holds. In fact, we get by Lemma 4.22 the stricter statement

$$\mathbf{K}(d)(\sigma(\theta), \sigma'(\theta)) \leq \sum_{x \in \mathcal{V}_s} (d(\sigma(x), \sigma'(x)) \cdot \mathrm{Var}(x, \theta)) + \sum_{\mu \in \mathcal{V}_d} (\mathbf{K}(d)(\sigma(\mu), \sigma'(\mu)) \cdot \mathrm{Var}(\mu, \theta)$$

which implies equation 4.13 since

$$\begin{aligned}
&\lambda \cdot \mathbf{K}(d)(\sigma(\theta), \sigma'(\theta)) \\
&\leq \sum_{x \in \mathcal{V}_s} \lambda \cdot d(\sigma(x), \sigma'(x)) \cdot \mathrm{Var}(x, \theta) + \sum_{\mu \in \mathcal{V}_d} \lambda \cdot \mathbf{K}(d)(\sigma(\mu), \sigma'(\mu)) \cdot \mathrm{Var}(\mu, \theta) \\
&= \sum_{i \in I} \left(\lambda \cdot d(\sigma(x_i), \sigma'(x_i)) \cdot \mathrm{Var}(x_i, \theta) + \sum_{\mu \in \mathrm{der}(r, x_i)} \lambda \cdot \mathbf{K}(d)(\sigma(\mu), \sigma'(\mu)) \cdot \mathrm{Var}(\mu, \theta) \right) \\
&\leq \sum_{i \in I} \left(\lambda \cdot d(\sigma(x_i), \sigma'(x_i)) \cdot \mathrm{Var}(x_i, \theta) + \sum_{\mu \in \mathrm{der}(r, x_i)} d(\sigma(x_i), \sigma'(x_i)) \cdot \mathrm{Var}(\mu, \theta) \right) \\
&= \sum_{i \in I} d(\sigma(x_i), \sigma'(x_i)) \left(\lambda \cdot \mathrm{Var}(x_i, \theta) + \sum_{\mu \in \mathrm{der}(r, x_i)} \mathrm{Var}(\mu, \theta) \right) \\
&= \sum_{i \in I} d(\sigma(x_i), \sigma'(x_i)) \cdot \mathsf{copy}(r, \lambda, x_i) \\
&\leq \sum_{i \in I} d(\sigma(x_i), \sigma'(x_i)) \cdot L(f)
\end{aligned}$$

$$
\begin{aligned}
&= L(f) \sum_{i \in I} d(\sigma(x_i), \sigma'(x_i)) \\
&= d(\sigma(f(x_1, \ldots, x_n)), \sigma'(f(x_1, \ldots, x_n)))
\end{aligned}
$$

with step 2 by the fact that the only variables in the target θ of the PGSOS rules r are source variables and their derivatives, step 3 by the fact that $\sigma(x_i)$ and $\sigma'(x_i)$ satisfy the λ-bisimulation metric transfer condition, step 5 by definition of copy, step 6 by property $\text{copy}(r, \lambda, x_i) \leq L(f)$ satisfied by the (L, λ)-Lipschitz rule r, and step 8 from the assumption $d(\underline{\sigma}(t), \underline{\sigma'}(t)) = L(f) \sum_{i=1}^{n} d(\underline{\sigma}(t_i), \underline{\sigma'}(t_i))$ and the equalities $\underline{\sigma}(t_i) = \sigma(x_i)$, $\underline{\sigma'}(t_i) = \sigma'(x_i)$, $\underline{\sigma}(t) = \sigma(f(x_1, \ldots, x_n))$ and $\underline{\sigma'}(t) = \sigma'(f(x_1, \ldots, x_n))$.

We conclude by observing that the transition $\sigma(f(x_1, \ldots, x_n)) \xrightarrow{a} \sigma(\text{trgt}(r))$ derived from the f-defining rule r can be mimicked by $\sigma'(f(x_1, \ldots, x_n)) \xrightarrow{a} \sigma'(\text{trgt}(r))$ (derived by the same r) such that the metric bisimulation transfer condition

$$
\begin{aligned}
&\lambda \, \mathbf{K}(d)(\sigma(\text{trgt}(r)), \sigma'(\text{trgt}(r))) \\
\leq &d(\sigma(f(x_1, \ldots, x_n)), \sigma'(f(x_1, \ldots, x_n))) \\
= &L(f) \sum_{i=1}^{n} \mathbf{d}(\sigma(x_i), \sigma'(x_i))
\end{aligned}
$$

holds. Hence, operator f is $L(f)$-Lipschitz continuous. □

It follows that a (L, λ)-Lipschitz PTSS specifies a non-expansive operator f if $L(f) \leq 1$.

4.3.3 Lipschitz continuous process algebra operators

Theorem 4.19 allows us to determine which process algebra operators are non-expansive (1-Lipschitz continuous) by inspecting their respective specification rules and verifying that the rule constraints of Definition 4.18 are satisfied for a Lipschitz factor of at most 1.

Corollary 4.23. *All standard non-recursive process algebra operators (specified in Figures 4.2 and 4.3), i.e.*

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

(and the non-extensive operators of Corollary 4.10) are non-expansive w.r.t. λ-bisimilarity metric $\mathbf{d}$ *for any* $\lambda \in (0, 1]$.

Proof. Let $P_1 = (\Sigma_1, A, R_1)$ be the PTSS consisting of all rules of Figure 4.2. Let $P_2 = (\Sigma_2, A, R_2)$ be the PTSS extending P_1 by the rules of Figure 4.3. Let L be defined as $L(f) = 1$ for all operators specified in Figure 4.3. It can be easily checked that the constraints in Definition 4.18 are satisfied. The thesis follows then directly by Theorem 4.19. □

This result is similar to Theorem 3.8. However, just like for non-extensiveness, we derive this result now by simple inspection of the structural properties of the specification rules.

On the other hand, the rules in Figure 4.4 specifying recursive process algebra operators do not admit any $L \in \mathscr{L}_\Sigma$ that assigns value 1 (for non-expansiveness) to any of its operators (except single iteration $\_^1$ and single replication $!^1\_$). It is easy to show that all those operators have a Lipschitz factor that is strictly greater than 1. As an example, we consider the finite iteration operator and show that the finite iteration operator with at least two iterations is not non-expansive.

Example 4.24. Consider the term $t = x^n$ and let σ_1 and σ_2 be the closed substitutions defined by $\sigma_1(x) = a.([1-\epsilon/\lambda]\varepsilon \oplus [\epsilon/\lambda]0)$ and $\sigma_2(x) = a.\varepsilon$, with any fixed $\epsilon \in (0,1)$ and $n > 1$. We have $\mathbf{d}(\sigma_1(x), \sigma_2(x)) = \epsilon$. Then by Proposition 3.9.a and Proposition 3.10 we get $\mathbf{d}(\sigma_1(t), \sigma_2(t)) = \epsilon(1-(\lambda-\epsilon)^n)/(1-(\lambda-\epsilon)) > \epsilon = \mathbf{d}(\sigma_1(x), \sigma_2(x))$, thus confirming that these arguments $\sigma_1(x)$ and $\sigma_2(x)$ are witnesses of the violation of the non-expansiveness condition for the n-iteration operator.

We consider now the specification of the finite iteration operator $\_^n$. First, observe that $L(\_^0) = 0$ since no rule specifies the operator $\_^0$. By induction, we get $L(\_^{n+1}) = 1 + \lambda L(\_^n)$. Since $\lambda > 0$, we conclude $L(\_^n) > 1$ for all $n \geq 2$, thus confirming that no Lipschitz factor $L \in \mathscr{L}_\Sigma$ with $L(\_^n) \leq 1$ gives rise to a (L,λ)-consistent PTSS for any $\lambda \in (0,1]$.

Theorem 4.19 allows us also to determine which process algebra operators are Lipschitz continuous by inspecting their respective specification rules and verifying that the rule constraints of Definition 4.18 are satisfied.

Corollary 4.25. *Consider the recursive process algebra operators of CCS and CSP specified in Figure 4.4. Then:*

1. *All finitely bounded recursive process algebra operators are Lipschitz continuous w.r.t. λ-bisimilarity metric $\mathbf{d}$ for any $\lambda \in (0,1]$.*

2. *All unbounded recursive process algebra operators are Lipschitz continuous w.r.t. λ-bisimilarity metric $\mathbf{d}$ for any $\lambda \in (0,1)$.*

Proof. Let $P_1 = (\Sigma_1, A, R_1)$ be the PTSS consisting of all rules of Figure 4.2. Let $P_2 = (\Sigma_2, A, R_2)$ be the PTSS extending P_1 by the rules of Figures 4.3 and 4.4. Let L be defined as $L(f) = 1$ for all operators specified in Figure 4.3. Let $L(\_^0) = L(!^0) = 0$. For the remaining operators we define $L(\_^n) = L(!^n) = \sum_{i=0}^{n-1} \lambda^i = (1-\lambda^n)/(1-\lambda)$, if $\lambda \in (0,1)$, and $L(\_^n) = L(!^n) = \sum_{i=0}^{n-1} \lambda^i = n$, if $\lambda = 1$. Then we define $L(\_^\omega) = L(\_^*\_) = \sum_{i=0}^{\infty} \lambda^i$ which is $L(\_^\omega) = L(\_^*\_) = \infty$ if $\lambda = 1$ and $L(\_^\omega) = L(\_^*\_) = 1/(1-\lambda)$ if $\lambda \in (0,1)$. Additionally, we define $L(\_^{*_p}\_) = L(\_^*\_)$ and $L(!_p\_) = 1/(p\lambda)$. It can be easily checked that the constraints in Definition 4.18 are satisfied. Cases 1 and 2 can now easily be verified by analyzing which operators f have $L(f) < \infty$. The thesis follows then directly by Theorem 4.19. □

This result is similar to Theorems 3.13 and 3.14. Note however, just as for non-extensiveness and non-expansiveness, that we derive this result now by simple inspection of the structural properties of the respective specifications. The unbounded recursive

process algebra operators are not Lipschitz continuous w.r.t. undiscounted bisimilarity metric.

4.3.4 Distance between Lipschitz continuous terms

We call a term t Lipschitz continuous if all operators used in t are Lipschitz continuous. Similar to non-extensive terms we can define an upper bound on the distance between closed instances of Lipschitz continuous terms. As before we will exploit the compositionality properties (i.e. Lipschitz factors) of each operator. The distance between closed instances of some Lipschitz continuous term t is the distance between the instances of the variables of t multiplied by the Lipschitz factors of the operators applied on top of the variables (expressed by the refined definition of Var provided in Equation 4.10).

Proposition 4.26. *Let P_1, P_2, L as in notation 4.17 and $P = (\Sigma', A, R)$ be any PGSOS PTSS with $P_2 \sqsubseteq P$. Then for any term $t \in \mathbb{T}(\Sigma_2)$ we have*

$$\mathbf{d}(\sigma(t), \sigma'(t)) \leq \sum_{x \in \mathcal{V}_s} \mathrm{Var}(x, t) \cdot \mathbf{d}(\sigma(x), \sigma'(x))$$

for all closed substitutions $\sigma, \sigma' : \mathcal{V} \to T(\Sigma')$.

Proof. Follows directly from Lemma 4.21. □

Note that Proposition 4.26 resembles Proposition 4.12 used for non-extensive contexts. Note again that the variables in t can be instantiated by arbitrary terms in $\mathsf{T}(\Sigma')$ (not necessarily built of Lipschitz continuous operators in Σ_2).

4.4 q-non-extensive operators

The specification format for Lipschitz continuous operators developed in Section 4.3 allows now also to define a specification format for q-non-extensive operators with $q \in (1, \infty)$. We start with analyzing which rule patterns specify and which do not specify q-non-extensive operators. Since ∞-non-extensiveness and q-non-extensiveness coincide for unary operators, the analysis of Example 4.2 in Section 4.2 (∞-non-extensiveness unary operators) carries over to q-non-extensiveness. It remains to explore binary operators.

Example 4.27. We consider the same rules specifying the binary operator f and the processes s_1, s_2, t_1, t_2 as in Example 4.3 in Section 4.2. However, we consider now any arbitrary fixed $q \in (1, \infty)$. Since ∞-non-extensiveness implies q-non-extensiveness, we revisit only those cases of Example 4.3 that may define operators that are not ∞-non-extensive.

Consider $\theta = \mu + \nu$. We already argued in Example 4.3 that this specification of f is not ∞-non extensive, whereas it follows by Theorem 4.19 that f is 1-non-extensive (which is 1-Lipschitz continuous). In Example 4.3 we argued that $\mathbf{d}(f(s_1, s_2), f(t_1, t_2)) = \lambda(1-(1-\epsilon_1)(1-\epsilon_2))$. The pairs of terms (s_1, s_2) and (t_1, t_2) are witnesses for the violation of the q-non-extensiveness condition, since for all $q > 1$ we have $\mathbf{d}(f(s_1, s_2), f(t_1, t_2)) =$

$\lambda(1-(1-\epsilon_1)(1-\epsilon_2)) > \lambda\epsilon_1 + \lambda\epsilon_2 > ((\lambda\epsilon_1)^q + (\lambda\epsilon_2)^q)^{\frac{1}{q}} = ((\mathbf{d}(s_1,t_1))^q + (\mathbf{d}(s_2,t_2))^q)^{\frac{1}{q}}$. Hence, this specification of operator f is not q-non-extensive for any $q \in (1,\infty)$.

Consider $\theta = (\mu + \nu) \oplus_r \delta(0)$ for some fixed $r \in (0,1)$. In Example 4.3 we derived $\mathbf{d}(f(s_1,s_2), f(t_1,t_2)) = r \cdot \lambda(1-(1-\epsilon_1)(1-\epsilon_2)) \leq r \cdot (\lambda\epsilon_1 + \lambda\epsilon_2)$. The q-non-extensiveness condition is then given if $r^q(\lambda\epsilon_1 + \lambda\epsilon_2)^q \leq (\lambda\epsilon_1)^q + (\lambda\epsilon_2)^q = (\mathbf{d}(s_1,t_1))^q + (\mathbf{d}(s_2,t_2))^q$. Since $r^q(\lambda\epsilon_1 + \lambda\epsilon_2)^q \leq r^q \cdot 2^{q-1}((\lambda\epsilon_1)^q + (\lambda\epsilon_2)^q)$ (Hölder inequality), the q-non-extensiveness condition is satisfied if $r^q \cdot 2^{q-1}((\lambda\epsilon_1)^q + (\lambda\epsilon_2)^q) \leq (\lambda\epsilon_1)^q + (\lambda\epsilon_2)^q$. This is given if $r^q \cdot 2^{q-1} \leq 1$, i.e. if $r \leq 2^{(1/q)-1}$. Theorem 4.29 will confirm that this specification of operator f is q-non-extensive if the probabilistic choice in the rule target is $r \leq 2^{(1/q)-1}$.

Consider $\theta = (\mu + \mu) \oplus_r \delta(0)$ for some fixed $r \in (0,1)$. Then $\mathbf{d}(f(s_1,s_2), f(t_1,t_2)) = r \cdot \lambda \cdot (1-(1-\epsilon_1)^2) \leq 2r\lambda\epsilon_1$. The q-non-extensiveness condition is given if $(2r\lambda\epsilon_1)^q \leq (\lambda\epsilon_1)^q + (\lambda\epsilon_2)^q = (\mathbf{d}(s_1,t_1))^q + (\mathbf{d}(s_2,t_2))^q$. This inequality holds if $r \leq 0.5$ by using $(2r\lambda\epsilon_1)^q \leq (\lambda\epsilon_1)^q \leq (\lambda\epsilon_1)^q + (\lambda\epsilon_2)^q$. Note that the constraint $r \leq 0.5$ on probability r is independent of the norm q. If $\theta = (\mu + \mu + \mu) \oplus_r \delta(0)$, then with similar arguments we get that the specified operator f is q-non-extensive if $r \leq 1/3$ (for any $q \in (1,\infty)$). Theorem 4.29 will confirm that the specified operators are q-non-extensive whenever the constraints on the probabilistic choice r are given.

Consider $\theta = (\delta(x) + \nu) \oplus_r \delta(0)$ for some fixed $r \in (0,1)$. In Example 4.3 we argued that $\mathbf{d}(f(s_1,s_2), f(t_1,t_2)) = r \cdot \lambda((1-\epsilon_2)\lambda\epsilon_1 + \epsilon_2)$. The q-non-extensiveness condition $(r \cdot \lambda((1-\epsilon_2)\lambda\epsilon_1 + \epsilon_2))^q \leq (\lambda\epsilon_1)^q + (\lambda\epsilon_2)^q$ is given for $r \leq 2^{(1/q)-1}$ since $(r \cdot \lambda((1-\epsilon_2)\lambda\epsilon_1 + \epsilon_2))^q \leq r^q 2^{q-1}((1-\epsilon_2)^q(\lambda^2\epsilon_1)^q + (\lambda\epsilon_2)^q) \leq r^q 2^{q-1}((\lambda\epsilon_1)^q + (\lambda\epsilon_2)^q) \leq$ (by $r \leq 2^{(1/q)-1}$) $(\lambda\epsilon_1)^q + (\lambda\epsilon_2)^q$. Theorem 4.29 will confirm that this specification of f is q-non-extensive whenever $r \leq 2^{(1/q)-1}$.

In essence, Example 4.27 shows that a binary operator f is q-non-extensive if the total number of instances of t_1, t_2 that are spawned along the evolution of the composed process $f(t_1,t_2)$, weighted by the probability of their realization, is at most $2^{(1/q)-1}$.

The q-norm is related to the 1-norm (which coincides with 1-Lipschitz continuity) by $n^{(1/q)-1}\sum_{i=1}^n d(t_i,t_i') \leq \left(\sum_{i=1}^n d(t_i,t_i')^q\right)^{1/q}$ (by Hölder inequality). Hence, each n-ary operator that is $n^{(1/q)-1}$-Lipschitz continuous is also q-non-extensive.

Definition 4.28 (q-non-extensiveness format)**.** Let P_1, P_2, L as in notation 4.17 and assume that P_2 is a (L,λ)-*Lipschitz PTSS*. We say that a rule $r \in R_2$ specifying an n-ary operator is a (L,λ,q)-*non-extensive rule* if

$$\mathsf{copy}(r,\lambda,x_i) \leq n^{(1/q)-1}$$

for all source variables $x_i \in \{x_1,\ldots,x_n\}$ of r.

Theorem 4.29. *Let $P = (\Sigma, A, R)$ be a (L,λ)-Lipschitz PTSS. Then, each n-ary operator $f \in \Sigma$ with $L(f) \leq n^{(1/q)-1}$ for some $q \in [1,\infty]$ is q-non-extensive.*

Proof. Directly by Theorem 4.19 and $n^{(1/q)-1}\sum_{i=1}^n d(t_i,t_i') \leq \left(\sum_{i=1}^n d(t_i,t_i')^q\right)^{1/q}$ (by Hölder inequality). □

We note that q-non-extensiveness is mainly of theoretical interest. The practically important case of ∞-non-extensiveness was already discussed in Section 4.2 and 1-non-extensiveness (which is 1-Lipschitz continuity) was discussed in Section 4.3.

4.5 Uniformly continuous operators

Uniform continuity is the most general compositionality property considered and hence will allow also for the largest class of possible specifications. A uniformly continuous operator ensures that a small variance in the behavior of a system component leads to a bounded small variance in the behavior of the composed system.

Definition 4.30 (Modulus of continuity)**.** Let $P = (\Sigma, A, R)$ be a PTSS, $f \in \Sigma$ some n-ary operator and d any 1-bounded pseudometric on $\mathsf{T}(\Sigma)$. A mapping $\omega\colon [0,1]^n \to [0,1]$ is an *upper bound on the distance between f-composed terms w.r.t. d* if

$$d(f(s_1,\ldots,s_n), f(t_1,\ldots,t_n)) \leq \omega(d(s_1,t_1),\ldots,d(s_n,t_n))$$

for all $s_i, t_i \in \mathsf{T}(\Sigma)$. An upper bound ω of f w.r.t. d is a *modulus of continuity of f w.r.t. d* if ω is continuous at $(0,\ldots,0)$, i.e. $\lim_{(\epsilon_1,\ldots,\epsilon_n)\to(0,\ldots,0)} \omega(\epsilon_1,\ldots,\epsilon_n) = \omega(0,\ldots,0)$, and $\omega(0,\ldots,0) = 0$.

An operator is uniformly continuous if this operator admits any modulus of continuity. Intuitively, a uniformly continuous binary operator f ensures that for any non-zero bisimulation distance ϵ (understood as the admissible tolerance from the operational behavior of the composed process $f(p_1,p_2)$) there are non-zero bisimulation distances δ_1 and δ_2 (understood as the admissible tolerances from the operational behavior of the processes p_1 and p_2) s.t. the distance between the composed processes $f(p_1,p_2)$ and $f(p_1',p_2')$ is at most ϵ whenever the component p_1' (resp. p_2') is in distance of at most δ_1 from p_1 (resp. at most δ_2 from p_2). More complex recursion patterns such as the fork operator of operating systems [BIM95] are only uniformly continuous (but not Lipschitz continuous, cf. Example 4.34 below).

Definition 4.31 (Uniformly continuous operator)**.** Let $P = (\Sigma, A, R)$ be a PTSS and d any 1-bounded pseudometric on $\mathsf{T}(\Sigma)$. We say that an operator $f \in \Sigma$ is

1. *uniformly continuous w.r.t. d* if f admits some modulus of continuity w.r.t. d,
2. *L-Lipschitz continuous w.r.t. d* with $L \in \mathbb{R}_{\geq 0}$ if $\omega(\epsilon_1,\ldots,\epsilon_n) = L\sum_{i=1}^n \epsilon_i$ is a modulus of continuity of f w.r.t. d,
3. *Lipschitz continuous w.r.t. d* if f is L-Lipschitz continuous w.r.t. d for some $L \in \mathbb{R}_{\geq 0}$, and
4. *q-non-extensive w.r.t. d* with $q \in [1,\infty]$ if $\omega(\epsilon_1,\ldots,\epsilon_n) = (\sum_{i=1}^n \epsilon_i^q)^{1/q}$, if $q < \infty$, and $\omega(\epsilon_1,\ldots,\epsilon_n) = \max_{i=1}^n \epsilon_i$, if $q = \infty$, is a modulus of continuity of f w.r.t. d.

Moreover, f is *non-expansive w.r.t. d* if f is 1-Lipschitz continuous w.r.t. d, and f is *non-extensive w.r.t. d* if f is ∞-non-extensive w.r.t. d.

Note that Definition 4.31.1 resembles Definition 3.12 of the former chapter (notion of modulus of continuity captures that for each $\epsilon > 0$ there exists some $(\delta_1,\ldots,\delta_n) > 0$), Definition 4.31.2 resembles earlier Definition 4.13 and Definition 4.31.4 resembles earlier Definition 4.1.

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 4.32. *Let $P = (\Sigma, A, R)$ be a PTSS and $s, t \in T(\Sigma)$ 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 $(\mathbf{d} - \epsilon)\colon \mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma) \to [0, \epsilon]$ 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:

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

by using the properties

$$\begin{aligned}
\mathbf{K}(d) &\sqsupseteq \mathbf{K}(d') \qquad \text{if } d \sqsupseteq d'\\
\mathbf{H}(d) &\sqsupseteq \mathbf{H}(d') \qquad \text{if } d \sqsupseteq d'\\
\mathbf{K}(d - \epsilon)(\pi, \pi') &\geq \mathbf{K}(d)(\pi, \pi') - \epsilon\\
\mathbf{H}(d - \epsilon)(\pi, \pi') &\geq \mathbf{H}(d)(\pi, \pi') - \epsilon
\end{aligned} \tag{4.14}$$

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 $\mathbf{d}(s,t) = \sup_{a \in A}\{\mathbf{H}(\lambda \cdot \mathbf{K}(\mathbf{d}))(der(s,a), der(t,a))\}$ applied in step 4, and properties of Equation 4.14 applied in steps 2 and 3. □

A fundamental insight that we will use later to define the specification format for uniform continuity is that an operator is uniformly continuous w.r.t. the bisimilarity metric $\mathbf{d}$ if this operator is Lipschitz continuous w.r.t. all up-to-k bisimilarity metrics $\mathbf{d}_k$. This will allows us later to specify uniformly continuous operators by specifying the Lipschitz factors of these operators for each up-to-k bisimulation distances.

Theorem 4.33. *Let $P = (\Sigma, A, R)$ be a PTSS and $\lambda < 1$. If an operator $f \in \Sigma$ is Lipschitz continuous w.r.t. $\mathbf{d}_k$ for each $k \in N$, then f is uniformly continuous w.r.t. $\mathbf{d}$.*

Proof. Assume that $f \in \Sigma$ is any n-ary operator. We will provide a modulus of continuity of f w.r.t. $\mathbf{d}$. Let $L_k \in \mathbb{R}_{\geq 0}$ be the Lipschitz factor for f w.r.t. $\mathbf{d}_k$, i.e.

$$\omega_k(\epsilon_1, \ldots, \epsilon_n) = L_k \sum_{i=1}^{n} \epsilon_i$$

is a modulus of continuity of f w.r.t. $\mathbf{d}_k$. Together with Proposition 4.32 and property $\mathbf{d}_k \sqsubseteq \mathbf{d}$ we get

$$\mathbf{d}(f(s_1, \ldots, s_n), f(t_1, \ldots, t_n)) \leq L_k \sum_{i=1}^{n} \mathbf{d}(s_i, t_i) + \lambda^k \tag{4.15}$$

for all $k \in \mathbb{N}$. Let $\omega\colon \mathbb{R}^n \to \mathbb{R}$ be the mapping defined by

$$\omega(\epsilon_1, \ldots, \epsilon_n) = \inf_{k\in\mathbb{N}} \left(L_k \sum_{i=1}^{n} \epsilon_i + \lambda^k \right).$$

From Equation 4.15 it is clear that ω is an upper bound on the distance between f-composed terms w.r.t. $\mathbf{d}$. Furthermore, $\omega(0, \ldots, 0) = 0$. To conclude that ω is a modulus of continuity, it remains to show that ω is continuous at $(0, \ldots, 0)$. Assume any $\delta \in (0, 1]$. Since $\lambda < 1$, there is some $m \in \mathbb{N}$ s.t. $\lambda^m < \delta$. Now, for any $(\epsilon_1, \ldots, \epsilon_n) \in (0, 1]^n$ with

$$\epsilon_i < \frac{\delta - \lambda^m}{n \cdot L_m}$$

we get

$$\begin{aligned}
&\omega(\epsilon_1, \ldots, \epsilon_n) \\
&= \inf_{k\in\mathbb{N}} \left(L_k \sum_{i=1}^{n} \epsilon_i + \lambda^k \right) \\
&\leq \inf_{k\in\mathbb{N}} \left(L_k \sum_{i=1}^{n} \frac{\delta - \lambda^m}{n \cdot L_m} + \lambda^k \right) \\
&\leq \inf_{k\in\mathbb{N}} \left(L_m \sum_{i=1}^{n} \frac{\delta - \lambda^m}{n \cdot L_m} + \lambda^k \right) \\
&\leq \inf_{k\in\mathbb{N}} \left(\delta - \lambda^m + \lambda^k \right) \\
&< \delta
\end{aligned}$$

Hence, ω is continuous at $(0, \ldots, 0)$. Thus, ω is a modulus of continuity of f w.r.t. $\mathbf{d}$. We conclude that f is uniformly continuous w.r.t. $\mathbf{d}$. □

4.5.1 Analysis of uniformly continuous operators

We analyze now the structural patterns of SOS rules that define uniformly continuous operators and give representative examples of rules that specify operators that are not uniformly continuous. Moreover, we derive from the structural properties of the rules the moduli of continuity of the specified operators.

Examples 4.14–4.16 in Section 4.3 showed that the number of process replications, weighted by the probability of their realization, and weighted by the discount factor if processes are delayed, determines the Lipschitz factor of the operator. Furthermore, the examples showed that if the number of recurring process replications is finitely bounded, then the specified operator is Lipschitz continuous.

Now we will analyze which rule patterns specify and which do not specify uniformly continuous operators.

Example 4.34 (Uniformly continuous operators)**.** We analyze now the fork operation of operating systems specified by the copy operator of [BIM95; FGW12] with the rules

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

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.

First, we show that the copy operator is not Lipschitz continuous. Formally, for any $L \in \mathbb{R}_{\geq 0}$, we show that $\mathbf{d}(\mathsf{cp}(s), \mathsf{cp}(t)) > L\mathbf{d}(s,t)$ for some CCS processes 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.

However, Proposition 4.40 below (exploiting Proposition 4.32) will confirm that if $\lambda < 1$ then $\omega(\epsilon) = \inf_{k\in\mathbb{N}}(2^k\epsilon + \lambda^k)$ is a (non-linear) modulus of continuity of the copy operator. Intuitively, the copy operator creates in k steps at most 2^k copies of the source process x, i.e. the copy operator is 2^k-Lipschitz continuous w.r.t. the up-to-k bisimulation metric $\mathbf{d}_k$. Thus, the copy operator is uniformly continuous (Theorem 4.33).

In essence, Example 4.34 shows that an operator is uniformly continuous if in each step only finitely many process copies are spawned.

Example 4.35 (Non-uniformly continuous operators)**.** Assume $A = \{a_k \mid k \in \mathbb{N}\}$. Consider the unary operators f and g specified by the following rules for all $k \in \mathbb{N}$:

$$\frac{x \xrightarrow{a_k} \mu}{f(x) \xrightarrow{a_k} \underbrace{\mu \mid \ldots \mid \mu}_{k\text{—times}}} \qquad \frac{}{g(x) \xrightarrow{a_k} \delta(\underbrace{h(\ldots h(}_{k\text{—times}} x))} \qquad \frac{x \xrightarrow{a_k} \mu}{h(x) \xrightarrow{a_k} \mu \mid \mu}$$

We will show that both operators are not uniformly continuous w.r.t. λ-bisimilarity metric $\mathbf{d}$ for any $\lambda \in (0,1]$. As arguments for the operators f and g we will use the terms $s_k = a_k.a_k.0$ and $t_{k,\epsilon} = a_k.([1-\epsilon]a_k.0 \oplus [\epsilon]0)$ with any $\epsilon \in (0,1)$. Clearly, $\mathbf{d}(s_k, t_{k,\epsilon}) = \lambda\epsilon$.

We start with operator f. We get $\mathbf{d}(f(s_k), f(t_{k,\epsilon})) = \lambda(1-(1-\epsilon)^k)$. Hence, if follows $\sup_{k\in\mathbb{N}} \mathbf{d}(f(s_k), f(t_{k,\epsilon})) = \sup_{k\in\mathbb{N}} \lambda(1-(1-\epsilon)^k) = \lambda$. The least upper bound on the distance between f-composed processes is $\omega(\epsilon) = \lambda$ if $\epsilon > 0$ and $\omega(0) = 0$. However, ω is not a modulus of continuity since it is not continuous at 0. Hence, operator f is not uniformly continuous.

We proceed with operator g. We get $\sup_{k\in\mathbb{N}} \mathbf{d}(g(s_k), g(t_{k,\epsilon})) = \sup_{k\in\mathbb{N}} \lambda^2(1-(1-\epsilon)^{2^k}) = \lambda^2$. Following the same line of argumentation as with operator f we conclude that operator g is not uniformly continuous.

In essence, Example 4.35 shows that an operator may be not uniformly continuous if it spawns in a single step an unbounded number of process copies.

4.5.2 Specification of uniformly continuous operators

We develop now a specification format that allows us to specify uniformly continuous operators.

We exploit Theorem 4.33 and specify uniformly continuous operators by defining suitable Lipschitz factors w.r.t. all up-to-k bisimilarity metrics. Technically, we will refine the Lipschitz factor assignments (Notation 4.17), and the operators Var and copy characterizing the process replication in rules (Equations 4.10 and 4.11) to consider the respective up-to-k notion of bisimilarity metric.

Notation 4.36. Let $P_1 = (\Sigma_1, A, R_1)$ be a λ-non-extensive PTSS, and $P_2 = (\Sigma_2, A, R_2)$ with $P_1 \sqsubseteq P_2$ be any PGSOS PTSS. Let $L\colon (\mathbb{N} \times \Sigma) \to \mathbb{R}^{\infty}_{\geq 0}$ with $\Sigma = \Sigma_2 \setminus \Sigma_1$ be a mapping[2] that assigns to each operator its respective up-to-k Lipschitz factors, or assigns ∞ if the operator is not Lipschitz continuous w.r.t. the up-to-k bisimulation metric. Let $\mathscr{L}_\Sigma$ be the set of all mappings $(\mathbb{N} \times \Sigma) \to \mathbb{R}^{\infty}_{\geq 0}$. We call $L \in \mathscr{L}_\Sigma$ a *Lipschitz factor assignment* (*LFA*, for short) for operators in Σ.

Intuitively, $L_k(f)$ gives either the Lipschitz factor of operator $f \in \Sigma$ w.r.t. λ-bisimilarity metric $\mathbf{d}_k$, or ∞ if f is not Lipschitz continuous w.r.t. $\mathbf{d}_k$.

We refine now the mapping Var (Equation 4.10) to incorporate the up-to-k Lipschitz factors by defining $\mathrm{Var}(\zeta, t, k)$ as the number of occurrences of variable ζ in t, weighted by the probability of their realization, and weighted by the up-to-k Lipschitz factors of the operators applied on top of ζ. Formally, $\mathrm{Var}\colon \mathscr{V} \times \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma) \times \mathbb{N} \to \mathbb{R}^{\infty}_{\geq 0}$ is defined by

$$\mathrm{Var}(\zeta, t, k) = \begin{cases} 1 & \text{if } t = \zeta \\ \max\limits_{i=1}^{n} \mathrm{Var}(\zeta, t_i, k) & \text{if } \left[\begin{array}{l} t = f(t_1, \ldots, t_n) \wedge \\ f \in \Sigma_1 \end{array}\right. \\ L_k(f) \sum\limits_{i=1}^{n} \mathrm{Var}(\zeta, t_i, k) & \text{if } \left[\begin{array}{l} t = f(t_1, \ldots, t_n) \wedge \\ f \in \Sigma_2 \setminus \Sigma_1 \end{array}\right. \\ 0 & \text{otherwise} \end{cases}$$

$$\mathrm{Var}(\zeta, \theta, k) = \begin{cases} 1 & \text{if } \theta = \zeta \\ \mathrm{Var}(\zeta, t, k) & \text{if } \theta = \delta(t) \\ \sum\limits_{i \in I} p_i \cdot \mathrm{Var}(\zeta, \theta_i, k) & \text{if } \theta = \sum\limits_{i \in I} p_i \theta_i \\ \max\limits_{i=1}^{n} \mathrm{Var}(\zeta, \theta_i, k) & \text{if } \left[\begin{array}{l} \theta = f(\theta_1, \ldots, \theta_n) \wedge \\ \zeta \in \mathscr{V}_s \wedge f \in \Sigma_1 \end{array}\right. \\ \sum\limits_{i=1}^{n} \mathrm{Var}(\zeta, \theta_i, k) & \text{if } \left[\begin{array}{l} \theta = f(\theta_1, \ldots, \theta_n) \wedge \\ \zeta \in \mathscr{V}_d \wedge f \in \Sigma_1 \end{array}\right. \\ L_k(f) \sum\limits_{i=1}^{n} \mathrm{Var}(\zeta, \theta_i, k) & \text{if } \left[\begin{array}{l} \theta = f(\theta_1, \ldots, \theta_n) \wedge \\ \zeta \in \mathscr{V}_s \wedge f \in \Sigma_2 \setminus \Sigma_1 \end{array}\right. \\ \overline{L_k(f)} \sum\limits_{i=1}^{n} \mathrm{Var}(\zeta, \theta_i, k) & \text{if } \left[\begin{array}{l} \theta = f(\theta_1, \ldots, \theta_n) \wedge \\ \zeta \in \mathscr{V}_d \wedge f \in \Sigma_2 \setminus \Sigma_1 \end{array}\right. \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

[2]We will write the first argument of L as subscript, i.e. $L_k(f)$ for $L(k, f)$, to align with the notation $\mathbf{d}_k$ of up-to-k bisimilarity metric. Hence, L_k denotes a function $\Sigma \to \mathbb{R}^{\infty}_{\geq 0}$ defined by $L_k(f) = L(k, f)$.

with $\overline{L_k(f)} = \max(L_k(f), 1)$.

We refine now the mapping copy (Equation 4.11) by using the refined definition of Var. The mapping $\mathsf{copy}\colon R \times (0,1] \times \mathscr{V}_s \times \mathbb{N} \to \mathbb{R}^{\infty}_{\geq 0}$ is defined as:

$$\mathsf{copy}(r, \lambda, x_i, k+1) = \lambda \cdot \mathrm{Var}(x_i, \mathrm{trgt}(r), k) + \sum_{\mu \in \mathrm{der}(r, x_i)} \mathrm{Var}(\mu, \mathrm{trgt}(r), k) \tag{4.17}$$

Intuitively, the expression $\mathsf{copy}(r, \lambda, x_i, k+1)$ describes how many copies of the source x_i and its derivatives $\mu \in \mathrm{der}(r, x_i)$ emerge after $k+1$ steps, with first transition specified by rule r.

An operator $f \in \Sigma$ is uniformly continuous if at most $L_k(f)$ instances of each source process or its derivatives (up to probabilistic weighting and discounting) evolve along the first k steps, for some $L_k(f) < \infty$.

Definition 4.37 (Uniform continuity format)**.** Let P_1, P_2, L as in notation 4.36. A rule $r \in R_2$ is a (L, λ)-*uniformly continuous* rule if

$$\mathsf{copy}(r, \lambda, x_i, k) \leq L_k(f)$$

for all source variables $x_i \in \{x_1, \ldots, x_n\}$ of r and for all $k \in \mathbb{N}$. We call P_2 a (L, λ)-*uniformly continuous PTSS* if all rules $r \in R_2$ are (L, λ)-uniformly continuous rules.

Uniformly continuous PTSSs specify uniformly continuous operators.

Theorem 4.38. *Let P_1, P_2, L as in notation 4.36 with P_2 a (L, λ)-uniformly continuous PTSS. Given any operator $f \in \Sigma$, if $L_k(f) < \infty$ for all $k \in \mathbb{N}$, then*

1. *f is $L_k(f)$-Lipschitz continuous w.r.t. $\mathbf{d}_k$ for any $k \in \mathbb{N}$, and*
2. *f is uniformly continuous if $\lambda < 1$.*

Proof. We start with Theorem 4.38.1. Given the PTSS $P_2 = (\Sigma_2, A, R_2)$ we will construct a PTSS P_2' such each term $t \in \mathsf{T}(\Sigma_2)$ has for any $k \in \mathbb{N}$ a corresponding term $t_k \in \mathsf{T}(\Sigma_2')$ that behaves for the first k steps as t and stops afterwards. Hence, if f is $L_k(f)$-Lipschitz continuous w.r.t. $\mathbf{d}_k$ in the induced model of P_2, then f_k is $L_k(f)$-Lipschitz continuous w.r.t. $\mathbf{d}$ in the induced model of P_2'. This reduces the proof of uniform continuity to the case of Lipschitz continuity (Theorem 4.33 and Theorem 4.19).

Given the signature $\Sigma_2 = (F, \mathsf{r})$ we define the signature $\Sigma_2' = (F', \mathsf{r}')$ by $F' = \{f_k \mid f \in F\}$ and $\mathsf{r}'(f_k) = \mathsf{r}(f)$. For each term $t \in \mathbb{T}(\Sigma_2) \cup \mathbb{DT}(\Sigma_2)$ we define by t_k the term in $\mathbb{T}(\Sigma_2') \cup \mathbb{DT}(\Sigma_2')$ obtained by replacing each function symbol f in t by f_k. Formally, for state terms we have $x_k = x$ and $(f(t^1, \ldots, t^n))_k = f_k(t^1_k, \ldots, t^n_k))$, and for distribution terms we have $\mu_k = \mu$, $\delta(t)_k = \delta(t_k)$, $(\sum_{i \in I} p^i \theta^i)_k = \sum_{i \in I} p^i \theta^i_k$, and $(f(\theta^n, \ldots, \theta^n))_k = f_k(\theta^n_k, \ldots, \theta^n_k)$. For distributions $\pi \in \Delta(\mathsf{T}(\Sigma_2))$ we define the distribution $\pi_k \in \Delta(\mathsf{T}(\Sigma_2'))$ defined by $\pi_k(t_k) = \pi(t)$. Given a rule $r \in R_2$ we define the rule r_{k+1} as a rule with the same premises, i.e. $\mathrm{prem}(r') = \mathrm{prem}(r)$, and $\mathrm{conc}(r_{k+1}) = f_{k+1}(x_1, \ldots, x_n) \xrightarrow{a} \theta_k$ if $\mathrm{conc}(r) = f(x_1, \ldots, x_n) \xrightarrow{a} \theta$. Note that no rule specifies a transition for operators f_0. In other words, terms $f_0(t^1, \ldots, t^n)$ cannot perform any transition. Let R_2' denote the set of rules $R_2' = \{r_k \mid r \in R \text{ and } k \in \mathbb{N}\}$.

Consider the PTSS (Σ_2', A, R_2'). A term $t_k \in \mathsf{T}(\Sigma_2')$ behaves like t for the first k moves and then stops. Formally, we have that t_0 makes no move and $t_{k+1} \xrightarrow{a} \pi_k$ iff $t \xrightarrow{a} \pi$. It follows that $\mathbf{d}_k(s,t) = \mathbf{d}_k(s_k,t_k) = \mathbf{d}(s_k,t_k)$ for all $s,t \in \mathsf{T}(\Sigma_2)$. Therefore, to prove the proof obligation that f is $L_k(f)$-Lipschitz continuous w.r.t. $\mathbf{d}_k$ it is enough to prove that f_k is $L_k(f)$-Lipschitz continuous w.r.t. $\mathbf{d}$.

Let $L' \colon (\Sigma_2' \setminus \Sigma_1') \to \mathbb{R}_{\geq 0}^{\infty}$ such that $L'(f_k) = L_k(f)$ for all $f \in \Sigma_2 \setminus \Sigma_1$ and $k \in \mathbb{N}$. The proof obligation becomes now that f_k is $L'(f_k)$-Lipschitz continuous w.r.t. $\mathbf{d}$. For each rule $r \in R$ and $k \in \mathbb{N}$ we have $\mathsf{copy}(r,\lambda,x_i,k) = \mathsf{copy}(r_k,\lambda,x_i)$. Hence $\mathsf{copy}(r_k,\lambda,x_i) \leq L'(f_k)$ follows from $\mathsf{copy}(r,\lambda,x_i,k) \leq L_k(f)$. Finally, since $L'(f_k) < \infty$, by Theorem 4.19 we get that f_k is $L'(f_k)$-Lipschitz continuous w.r.t. $\mathbf{d}$. Hence, f is $L_k(f)$-Lipschitz continuous w.r.t. $\mathbf{d}_k$.

Then Theorem 4.38.2 follows directly from Theorem 4.38.1 and Theorem 4.33. □

Note that if $L_k(f) = L_{k'}(f)$ for all $k,k' \in \mathbb{N}$ then the up-to-k Lipschitz factor assignment $L \colon (\mathbb{N} \times (\Sigma_2 \setminus \Sigma_1)) \to \mathbb{R}_{\geq 0}^{\infty}$ (Notation 4.36) becomes a Lipschitz factor assignment $L \colon (\Sigma_2 \setminus \Sigma_1) \to \mathbb{R}_{\geq 0}^{\infty}$ (Notation 4.17). Then, the uniform continuity rule format (Definition 4.37) coincides with the Lipschitz continuity format (Definition 4.18). Hence the specified operator is Lipschitz continuous.

4.5.3 Uniformly continuous process algebra operators

Theorem 4.38 allows us to determine which process algebra operators are uniformly continuous by inspecting their respective specification rules and verifying that the rule constraints of Definition 4.37 are satisfied. We provide now an example that shows how to derive Lipschitz factor assignment that makes the specification uniformly continuous, and then how to determine the resp. modulus of continuity.

Corollary 4.39. *The copy operator* cp *is uniformly continuous w.r.t. λ-bisimilarity metric* $\mathbf{d}$ *for any* $\lambda \in (0,1)$.

Proof. Let $P = (\Sigma, A, R)$ be the PTSS specifying the synchronous parallel composition operator and the copy operator (Example 4.34). Let $L \in \mathscr{L}_{\Sigma}$ be defined as $L_k(|) = 1$ and $L_k(\mathsf{cp}) = 2^k$ for any $k \in \mathbb{N}$. Let r_1 and r_2 denote the rules for operator cp, namely

$$r_1 = \frac{x \xrightarrow{a} \mu}{\mathsf{cp}(x) \xrightarrow{a} \mu} \qquad r_2 = \frac{x \xrightarrow{l} \mu \quad x \xrightarrow{r} \nu}{\mathsf{cp}(x) \xrightarrow{s} \mathsf{cp}(\mu) \mid \mathsf{cp}(\nu)}$$

First, note that for each $k \in \mathbb{N}$ we have $\mathrm{Var}(\mu,\mu,k) = 1$ and $\mathrm{Var}(\mathsf{cp}(\mu) \mid \mathsf{cp}(\nu), \mu, k) = \mathrm{Var}(\mathsf{cp}(\mu) \mid \mathsf{cp}(\nu), \nu, k) = 2^k$. Then $\mathsf{copy}(r_1,\lambda,x,k+1) = 1$ and $\mathsf{copy}(r_2,\lambda,x,k+1) = 2^k + 2^k = 2^{k+1}$. Since $\sup_{r \in R_f} \mathsf{copy}(r,\lambda,x,k) \leq L_k(f)$ the PTSS P is (L,λ)-uniformly continuous. Thus, if $\lambda < 0$ we get that cp is uniformly continuous (Theorem 4.38.2). □

4.5.4 Distance between uniformly continuous terms

We call a term t uniformly continuous if all operators used in t are uniformly continuous. As before we define now an upper bound on the distance between closed instances of uniformly continuous terms.

Proposition 4.40. *Let P_1, P_2, L as in notation 4.36 and $P = (\Sigma', A, R')$ be any PGSOS PTSS with $P_2 \sqsubseteq P$. Then for any term $t \in \mathbb{T}(\Sigma_2)$ we have*

$$\mathbf{d}(\sigma(t), \sigma'(t)) \leq \inf_{k \in \mathbb{N}} \left(\sum_{x \in \mathcal{V}_s} \mathrm{Var}(x, t, k) \cdot \mathbf{d}(\sigma(x), \sigma'(x)) + \lambda^k \right)$$

for all closed substitutions $\sigma, \sigma' : \mathcal{V} \to T(\Sigma')$.

Proof. We use the same notation $\Sigma_2' = (F', \mathrm{r}')$ as in the proof of Theorem 4.38. Notice that $\mathrm{Var}(x, t_k) = \mathrm{Var}(x, t, k)$ for all $x \in \mathcal{V}_s$, $t \in \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)$ and $k \in \mathbb{N}$. Given a closed substitution $\sigma : \mathcal{V} \to \mathsf{T}(\Sigma')$ we define $\sigma_k(x) = \sigma(x)_k$, i.e. $\sigma_k(x) = t_k$ if $\sigma(x) = t$. Then

$$\begin{aligned}
&\mathbf{d}(\sigma(t), \sigma'(t)) \\
&\leq \inf_{k \in \mathbb{N}} \left(\mathbf{d}_k(\sigma(t), \sigma'(t)) + \lambda^k \right) && \text{(Proposition 4.32)} \\
&= \inf_{k \in \mathbb{N}} \left(\mathbf{d}((\sigma(t))_k, (\sigma'(t))_k) + \lambda^k \right) && \text{(see proof of Theorem 4.38)} \\
&= \inf_{k \in \mathbb{N}} \left(\mathbf{d}(\sigma_k(t_k), \sigma'_k(t_k)) + \lambda^k \right) \\
&\leq \inf_{k \in \mathbb{N}} \left(\left(\sum_{x \in \mathcal{V}_s} \mathrm{Var}(x, t_k) \cdot \mathbf{d}(\sigma_k(x), \sigma'_k(x)) \right) + \lambda^k \right) && \text{(Proposition 4.26)} \\
&= \inf_{k \in \mathbb{N}} \left(\left(\sum_{x \in \mathcal{V}_s} \mathrm{Var}(x, t, k) \cdot \mathbf{d}(\sigma(x), \sigma'(x)) \right) + \lambda^k \right) && (\mathrm{Var}(x, t_k) = \mathrm{Var}(x, t, k)).
\end{aligned}$$

□

4.6 Coinductive rule format characterization

In the former Sections 4.2–4.5 we developed syntactic compositionality results (SOS rule and specification formats) w.r.t. the compositionality properties of non-extensiveness, non-expansiveness, Lipschitz continuity and uniform continuity. In this section we develop now a theory to relate syntactic compositionality properties with semantic compositionality properties (moduli of continuity). This allows us not only to define for any modulus of continuity an expressive specification format (Theorem 4.54) but also to derive from any given specification the moduli of continuity of the specified operators (Theorem 4.58). Technically, the specification rules induce a function on the space of Lipschitz factor assignments such that the prefixed points of this function are consistent Lipschitz factor assignments, i.e. those that denote valid Lipschitz factors of each operator. Intuitively, the function describes the process replication along the specified transitions and the prefixed points satisfy a replication invariance condition (mimicking the bisimulation metric invariance condition). In other words, consistent Lipschitz factor assignments are those that are invariant along the transitions described by the rules.

4.6.1 Finite projection Lipschitz continuous operators

We will start by developing an alternative (coinductive) specification format to specify Lipschitz continuous and uniformly continuous operators. To keep the presentation simple

we consider only Lipschitz and uniformly continuous operators. It is straightforward to extend all developments in this section to include also non-extensive operators. For the remainder of this section we assume a PTSS $P = (\Sigma, A, R)$ and a Lipschitz factor assignments $L \in \mathscr{L}_\Sigma$, i.e. we assume Notation 4.36 with Σ_1 an empty signature and $\Sigma_2 = \Sigma$.

Proposition 4.41. *$(\mathscr{L}_\Sigma, \sqsubseteq)$ is a complete lattice.*

Proof. First, observe that $(\mathbb{R}^\infty_{\geq 0}, \leq)$ is a complete lattice. Let $\mathscr{M} \subseteq \mathscr{L}_\Sigma$ be an arbitrary set of Lipschitz factor assignments. The supremum of $\mathscr{M}$ is given by $(\sup \mathscr{M})_k(f) = \sup_{L \in \mathscr{M}} L_k(f)$, and the infimum of $\mathscr{M}$ is given by $(\inf \mathscr{M})_k(f) = \inf_{L \in \mathscr{M}} L_k(f)$. Hence, $(\mathscr{L}_\Sigma, \sqsubseteq)$ is a complete lattice. □

It is clear that the bottom element of $(\mathscr{L}_\Sigma, \sqsubseteq)$ is the LFA $0 \in \mathscr{L}_\Sigma$ given by $0_k(f) = 0$ for all $k \in \mathbb{N}$ and $f \in \Sigma$.

Definition 4.42 (Semantic consistency)**.** Let $L \in \mathscr{L}_\Sigma$ be a LFA and $k \in \mathbb{N}$. We call L *consistent with the up-to-k bisimilarity metric* $\mathbf{d}_k$ if

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

for all operators $f \in \Sigma$ and all terms $s_i, t_i \in \mathsf{T}(\Sigma)$. Furthermore, we call L *consistent with the bisimilarity metric* $\mathbf{d}$ if L is consistent with $\mathbf{d}_k$ for all $k \in \mathbb{N}$.

Hence $L \in \mathscr{L}_\Sigma$ is consistent with $\mathbf{d}_k$ if each operator f with $L_k(f) < \infty$ is $L_k(f)$-Lipschitz continuous w.r.t. $\mathbf{d}_k$.

We proceed by lifting LFAs from operators to terms.

Definition 4.43 (Lifting of a Lipschitz factor assignment)**.** Let $L \in \mathscr{L}_\Sigma$ be a LFA. The lifting of L is a *Lipschitz factor assignment on terms* given as the mapping $L : (\mathbb{N} \times (\mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)) \times \mathscr{V}) \to \mathbb{R}^\infty_{\geq 0}$ defined by:

$$L_k(t, \zeta) = \begin{cases} 1 & \text{if } t = \zeta \\ L_k(f) \sum_{i=1}^{n} L_k(t_i, \zeta) & \text{if } t = f(t_1, \ldots, t_n) \\ 0 & \text{otherwise} \end{cases}$$

$$L_k(\theta, \zeta) = \begin{cases} 1 & \text{if } \theta = \zeta \\ L_k(t, \zeta) & \text{if } \theta = \delta(t) \\ \sum_{i \in I} p_i \cdot L_k(\theta_i, \zeta) & \text{if } \theta = \sum_{i \in I} p_i \theta_i \\ L_k(f) \sum_{i=1}^{n} L_k(\theta_i, \zeta) & \text{if } \theta = f(\theta_1, \ldots, \theta_n) \text{ and } \zeta \in \mathscr{V}_s \\ \overline{L_k(f)} \sum_{i=1}^{n} L_k(\theta_i, \zeta) & \text{if } \theta = f(\theta_1, \ldots, \theta_n) \text{ and } \zeta \in \mathscr{V}_d \\ 0 & \text{otherwise} \end{cases}$$

with $\overline{L_k(f)} = \max(L_k(f), 1)$.

The Lipschitz factor of a state term arises from the functional composition of the Lipschitz moduli of continuity of the operators in the state term. This is also the case for distribution terms except for operators with $L_k(f) < 1$ (case 5 of $L_k(\theta, \zeta)$). As shown in Examples 4.2 and 4.3, if f has a modulus of continuity on state terms below 1-Lipschitz continuity (e.g. ∞-non-extensiveness), then the modulus of continuity of f on distribution terms is 1-Lipschitz continuity (but not smaller).

The lifting of a LFA preserves consistency.

Proposition 4.44. *Let $L \in \mathscr{L}_\Sigma$ be a LFA and $k \in \mathbb{N}$. If L is consistent with $\mathbf{d}_k$, then for all closed substitutions $\sigma_1, \sigma_2 \colon \mathscr{V} \to T(\Sigma) \cup DT(\Sigma)$ we get*

1. $\mathbf{d}_k(\sigma_1(t), \sigma_2(t)) \leq \sum_{x \in \mathscr{V}_s} L_k(t, x) \cdot \mathbf{d}_k(\sigma_1(x), \sigma_2(x))$ *for any state term $t \in \mathbb{T}(\Sigma)$*
2. $\mathbf{K}(\mathbf{d}_k)(\sigma_1(\theta), \sigma_2(\theta)) \leq \sum_{x \in \mathscr{V}_s} L_k(\theta, x) \cdot \mathbf{d}_k(\sigma_1(x), \sigma_2(x)) +$
   $\sum_{\mu \in \mathscr{V}_d} L_k(\theta, \mu) \cdot \mathbf{K}(\mathbf{d}_k)(\sigma_1(\mu), \sigma_2(\mu))$ *for any distribution term $\theta \in \mathbb{DT}(\Sigma)$.*

Proof. We start with showing Proposition 4.44.1. We reason by structural induction over t. The base case $t = x \in \mathscr{V}_s$ is immediate since $L_k(x, x) = 1$. Consider the inductive step $t = f(t_1, \ldots, t_n)$. We have

$$\begin{aligned}
&\mathbf{d}_k(\sigma_1(f(t_1, \ldots, t_n)), \sigma_2(f(t_1, \ldots, t_n))) \\
\leq &L_k(f) \sum_{i=1}^{n} \mathbf{d}_k(\sigma_1(t_i), \sigma_2(t_i)) \\
\leq &L_k(f) \sum_{i=1}^{n} \sum_{x \in \mathscr{V}_s} L_k(t_i, x) \cdot \mathbf{d}_k(\sigma_1(x), \sigma_2(x)) \\
= &\sum_{x \in \mathscr{V}_s} \left(L_k(f) \cdot \sum_{i=1}^{n} L_k(t_i, x) \right) \cdot \mathbf{d}_k(\sigma_1(x), \sigma_2(x)) \\
= &\sum_{x \in \mathscr{V}_s} L_k(f(t_1, \ldots, t_n), x) \cdot \mathbf{d}_k(\sigma_1(x), \sigma_2(x)).
\end{aligned}$$

with step 1 by consistency of L_k with $\mathbf{d}_k$, step 2 by the inductive hypothesis and step 4 by the definition of $L_k(f(t_1, \ldots, t_n), x)$.

We proceed by showing Proposition 4.44.2. We reason by structural induction over θ. The base case $\theta = \mu$ is immediate since $L_k(\mu, \mu) = 1$. The base case $\theta = \delta(t)$ follows directly from $\mathbf{K}(\mathbf{d}_k)(\sigma_1(\delta(t)), \sigma_2(\delta(t))) = \mathbf{d}_k(\sigma_1(t), \sigma_2(t))$, Proposition 4.44.1, and $L_k(\delta(t), x) = L_k(t, x)$.

Consider the inductive step $\theta = \sum_{i \in I} p_i \theta_i$. We have

$$\begin{aligned}
&\mathbf{K}(\mathbf{d}_k)\big(\sigma_1\big(\sum_{i \in I} p_i \theta_i\big), \sigma_2\big(\sum_{i \in I} p_i \theta_i\big)\big) \\
\leq &\sum_{i \in I} p_i \, \mathbf{K}(\mathbf{d}_k)(\sigma_1(\theta_i), \sigma_2(\theta_i)) \\
\leq &\sum_{i \in I} p_i \Big(\sum_{x \in \mathscr{V}_s} L_k(\theta_i, x) \cdot \mathbf{d}_k(\sigma_1(x), \sigma_2(x)) + \sum_{\mu \in \mathscr{V}_d} L_k(\theta_i, \mu) \cdot \mathbf{K}(\mathbf{d}_k)(\sigma_1(\mu), \sigma_2(\mu)) \Big)
\end{aligned}$$

$$= \sum_{x \in \mathcal{V}_s} L_k \left(\sum_{i \in I} p_i \theta_i, x \right) \cdot \mathbf{d}_k(\sigma(x), \sigma'(x)) + \sum_{\mu \in \mathcal{V}_d} L_k \left(\sum_{i \in I} p_i \theta_i, \mu \right) \cdot \mathbf{K}(\mathbf{d}_k)(\sigma_1(\mu), \sigma_2(\mu))$$

with step 1 by Proposition 2.32.3, step 2 by the inductive hypothesis, and step 3 by the definition of L_k.

Finally, consider the inductive step $\theta = f(\theta_1, \ldots, \theta_n)$. We have

$$\begin{aligned}
&\mathbf{K}(\mathbf{d}_k)(\sigma_1(f(\theta_1, \ldots, \theta_n)), \sigma_2(f(\theta_1, \ldots, \theta_n))) \\
\leq & L_k(f) \sum_{i=1}^{n} \mathbf{K}(\mathbf{d}_k)(\sigma_1(\theta_i), \sigma_2(\theta_i)) \\
\leq & L_k(f) \sum_{i=1}^{n} \left(\sum_{x \in \mathcal{V}_s} L_k(\theta_i, x) \cdot \mathbf{d}_k(\sigma_1(x), \sigma_2(x)) + \sum_{\mu \in \mathcal{V}_d} L_k(\theta_i, \mu) \cdot \mathbf{K}(\mathbf{d}_k)(\sigma_1(\mu), \sigma_2(\mu)) \right) \\
= & \sum_{x \in \mathcal{V}_s} \mathbf{d}_k(\sigma_1(x), \sigma_2(x)) \cdot L_k(f) \cdot \sum_{i=1}^{n} L_k(\theta_i, x) + \\
& \sum_{\mu \in \mathcal{V}_d} \mathbf{K}(\mathbf{d}_k)(\sigma_1(\mu), \sigma_2(\mu)) \cdot L_k(f) \cdot \sum_{i=1}^{n} L_k(\theta_i, \mu) \\
\leq & \sum_{x \in \mathcal{V}_s} \mathbf{d}_k(\sigma_1(x), \sigma_2(x)) \cdot L_k(f) \cdot \sum_{i=1}^{n} L_k(\theta_i, x) + \\
& \sum_{\mu \in \mathcal{V}_d} \mathbf{K}(\mathbf{d}_k)(\sigma_1(\mu), \sigma_2(\mu)) \cdot \overline{L_k(f)} \cdot \sum_{i=1}^{n} L_k(\theta_i, \mu) \\
= & \sum_{x \in \mathcal{V}_s} L_k(f(\theta_1, \ldots, \theta_n), x) \cdot \mathbf{d}_k(\sigma_1(x), \sigma_2(x)) + \\
& \sum_{\mu \in \mathcal{V}_d} L_k(f(\theta_1, \ldots, \theta_n), \mu) \cdot \mathbf{K}(\mathbf{d}_k)(\sigma_1(\mu), \sigma_2(\mu))
\end{aligned}$$

with step 1 by Corollary 2.34, step 2 by the inductive hypothesis, and step 5 by the definition of L_k. □

The rules R give rise to a mapping $R\colon \mathscr{L}_\Sigma \to \mathscr{L}_\Sigma$ with $R(L)$ defined as the Lipschitz factor assignment obtained by applying the rules of R to L.

Definition 4.45 (R-extension)**.** The *R-extension of LFAs* is the mapping[3] $R\colon \mathscr{L}_\Sigma \to \mathscr{L}_\Sigma$ defined by

$$\begin{aligned}
R(L)_0(f) &= 0 \\
R(L)_{k+1}(f) &= \sup_{r \in R_f} \max_{i=1}^{r(f)} \left(\lambda \cdot L_k(\mathrm{trgt}(r), x_i) + \sum_{\mu \in \mathrm{der}(r, x_i)} L_k(\mathrm{trgt}(r), \mu) \right)
\end{aligned}$$

for all $L \in \mathscr{L}_\Sigma$ and $f \in \Sigma$.

[3]The symbol R denotes both the set of rules of some specification and the R-extension mapping of LFAs induced by a set of rules R. The meaning of symbol R will always be clear from the application context.

Intuitively, the lifted LFA on terms (Definition 4.43) is obtained by structural induction over terms, while the R-extended LFA (Definition 4.45) is obtained by operational induction over rules. The R-extension of Lipschitz factor assignments preserves semantic consistency.

Proposition 4.46. *Let $L \in \mathscr{L}_\Sigma$ be a LFA and $k \in \mathbb{N}$. If L is consistent with $\mathbf{d}_k$, then $R(L)$ is consistent with $\mathbf{d}_{k+1}$.*

Proof. We need to show that

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

for all $s_i, t_i \in \mathsf{T}(\Sigma)$ and any $f \in \Sigma$, assuming that

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

for all $s_i, t_i \in \mathsf{T}(\Sigma)$ and any $f \in \Sigma$.

We will use the following notation. Given a closed substitution $\sigma\colon \mathscr{V} \to \mathsf{T}(\Sigma)\cup\mathsf{DT}(\Sigma)$ and a rule $r \in R$ we write $\sigma \models r$ if for all positive premises $x_i \xrightarrow{a_{i,k}} \mu_{i,k} \in \mathrm{pprem}(r)$ we have $\sigma(x_i) \xrightarrow{a_{i,k}} \sigma(\mu_{i,k}) \in \rightarrow$, and for all negative premises $x_i \xrightarrow{b_{i,l}} \in \mathrm{nprem}(r)$ we have that $\sigma(x_i) \xrightarrow{b_{i,l}} \pi \notin \rightarrow$ for all $\pi \in \Delta(\mathsf{T}(\Sigma))$. We write X_r for the set of source variables $\{x_1,\dots,x_n\}$ of r. We denote by $r(\sigma_1,\sigma_2)$ the set of all pairs of substitutions (σ_1',σ_2') that coincide on the source variables X_r with (σ_1,σ_2) and that satisfy the bisimulation transfer condition. Technically, $(\sigma_1',\sigma_2') \in r(\sigma_1,\sigma_2)$ if $\sigma_1' \models r$, $\sigma_2' \models r$, $\sigma_1' \restriction X_r = \sigma_1 \restriction X_r$, $\sigma_2' \restriction X_r = \sigma_2 \restriction X_r$, and for all positive premises $x_i \xrightarrow{a_{i,k}} \mu_{i,k} \in \mathrm{pprem}(r)$ we have $\lambda \cdot \mathbf{K}(\mathbf{d}_k)(\sigma_1'(\mu_{i,k}),\sigma_2'(\mu_{i,k})) \leq \mathbf{d}_{k+1}(\sigma_1'(x_i),\sigma_2'(x_i))$.

Let σ_1,σ_2 be the substitution defined by $\sigma_1(x_i) = s_i$, $\sigma_2(x_i) = t_i$, and identity for all other variables. Then

$$\begin{aligned}
&\mathbf{d}_{k+1}(f(s_1,\dots,s_n), f(t_1,\dots,t_n))\\
=&\sup_{a\in A}\{\mathbf{H}(\lambda\cdot\mathbf{K}(\mathbf{d}_k))(der(f(s_1,\dots,s_n),a), der(f(t_1,\dots,t_n),a))\}\\
\leq&\lambda \sup_{\substack{r\in R_f\\(\sigma_1',\sigma_2')\in r(\sigma_1,\sigma_2)}}\left\{\mathbf{K}(\mathbf{d}_k)(\sigma_1'(\mathrm{trgt}(r)),\sigma_2'(\mathrm{trgt}(r)))\right\}\\
\leq&\lambda \sup_{\substack{r\in R_f\\(\sigma_1',\sigma_2')\in r(\sigma_1,\sigma_2)}}\left\{\sum_{i=1}^{n}\left(\mathbf{d}_k(\sigma_1'(x_i),\sigma_2'(x_i))\cdot L_k(\mathrm{trgt}(r),x_i) + \right.\right.\\
&\qquad\left.\left.\sum_{\mu\in\mathrm{der}(r,x_i)}\mathbf{K}(\mathbf{d}_k)(\sigma_1'(\mu),\sigma_2'(\mu))\cdot L_k(\mathrm{trgt}(r),\mu)\right)\right\}\\
=&\sup_{\substack{r\in R_f\\(\sigma_1',\sigma_2')\in r(\sigma_1,\sigma_2)}}\left\{\sum_{i=1}^{n}\left(\mathbf{d}_k(\sigma_1'(x_i),\sigma_2'(x_i))\cdot\lambda\cdot L_k(\mathrm{trgt}(r),x_i) + \right.\right.
\end{aligned}$$

$$\left.\left.\sum_{\mu\in\mathrm{der}(r,x_i)} \lambda\cdot\mathbf{K}(\mathbf{d}_k)(\sigma_1'(\mu),\sigma_2'(\mu))\cdot L_k(\mathrm{trgt}(r),\mu)\right)\right\}$$

$$\leq \sup_{\substack{r\in R_f\\(\sigma_1',\sigma_2')\in r(\sigma_1,\sigma_2)}} \left\{\sum_{i=1}^{n}\left(\mathbf{d}_k(\sigma_1'(x_i),\sigma_2'(x_i))\cdot\lambda\cdot L_k(\mathrm{trgt}(r),x_i)+\right.\right.$$

$$\left.\left.\sum_{\mu\in\mathrm{der}(r,x_i)} \mathbf{d}_{k+1}(\sigma_1'(x_i),\sigma_2'(x_i))\cdot L_k(\mathrm{trgt}(r),\mu)\right)\right\}$$

$$\leq \sup_{\substack{r\in R_f\\(\sigma_1',\sigma_2')\in r(\sigma_1,\sigma_2)}} \left\{\sum_{i=1}^{n}\mathbf{d}_{k+1}(\sigma_1'(x_i),\sigma_2'(x_i))\left(\lambda\cdot L_k(\mathrm{trgt}(r),x_i)+\right.\right.$$

$$\left.\left.\sum_{\mu\in\mathrm{der}(r,x_i)} L_k(\mathrm{trgt}(r),\mu)\right)\right\}$$

$$\leq \sup_{\substack{r\in R_f\\(\sigma_1',\sigma_2')\in r(\sigma_1,\sigma_2)}} \left\{\max_{i=1}^{n}\left(\lambda\cdot L_k(\mathrm{trgt}(r),x_i)+\sum_{\mu\in\mathrm{der}(r,x_i)} L_k(\mathrm{trgt}(r),\mu)\right)\cdot\right.$$

$$\left.\sum_{i=1}^{n}\mathbf{d}_{k+1}(\sigma_1'(x_i),\sigma_2'(x_i))\right\}$$

$$= \sup_{r\in R_f}\left\{\max_{i=1}^{n}\left(\lambda\cdot L_k(\mathrm{trgt}(r),x_i)+\sum_{\mu\in\mathrm{der}(r,x_i)} L_k(\mathrm{trgt}(r),\mu)\right)\right\}\cdot$$

$$\sum_{i=1}^{n}\mathbf{d}_{k+1}(\sigma_1(x_i),\sigma_2(x_i))$$

$$=(R(L))_{k+1}(f)\sum_{i=1}^{n}\mathbf{d}_{k+1}(s_i,t_i)$$

where step 2 follows by the definition of functional **H**, step 3 follows by Proposition 4.44.2, step 5 follows from $(\sigma_1',\sigma_2')\in r(\sigma_1,\sigma_2)$ and the bisimulation transfer condition $\mathbf{H}(\lambda\cdot\mathbf{K}(\mathbf{d}_k))(\mathrm{der}(\sigma_1'(x_i),a),\mathrm{der}(\sigma_2'(x_i),a))\leq\mathbf{d}_{k+1}(\sigma_1'(x_i),\sigma_2'(x_i))$, and step 6 by $\mathbf{d}_k\sqsubseteq\mathbf{d}_{k+1}$. □

Corollary 4.47. *Let $L\in\mathscr{L}_\Sigma$ be a LFA. If L is consistent with $\mathbf{d}$, then $R(L)$ is consistent with $\mathbf{d}$.*

Proof. By definition, L is consistent with $\mathbf{d}$ iff L is consistent with $\mathbf{d}_k$ for all $k\in\mathbb{N}$. By Proposition 4.46 we get that $R(L)$ is consistent with $\mathbf{d}_k$ for all $k\in\mathbb{N}_{>0}$. Finally, all LFAs are clearly consistent with $\mathbf{d}_0=0$. Thus $R(L)$ is consistent with $\mathbf{d}_k$ for all $k\in\mathbb{N}$, i.e. $R(L)$ is consistent with $\mathbf{d}$. □

The R-extension mapping allows us to specify a canonical LFA given as the least fixed-point of R. Existence and uniqueness follow by the Knaster-Tarski theorem using that $(\mathscr{L}_\Sigma, \sqsubseteq)$ is a complete lattice (Proposition 4.41) and that R is monotone (Proposition 4.48). Since the bottom LFA $0 \in \mathscr{L}_\Sigma$ is consistent with $\mathbf{d}_0$ and R preserves consistency of LFAs (Proposition 4.46 and Corollary 4.47), we get that the canonical LFA is consistent with $\mathbf{d}$. The canonical LFA provides the least restricting syntactic requirements for the specified operators.

Proposition 4.48. *The R-extension mapping R is order-preserving on* $(\mathscr{L}_\Sigma, \sqsubseteq)$.

Proof. Assume arbitrary Lipschitz factor assignments $L, M \in \mathscr{L}_\Sigma$ with $L \sqsubseteq M$. We need to show $R(L) \sqsubseteq R(M)$, namely

$$\forall k \in \mathbb{N}.\ \forall f \in \Sigma.\ R(L)_k(f) \leq R(M)_k(f). \tag{4.18}$$

We proceed by induction over k. The base case $k = 0$ is trivial since $R(L)_0(f) = R(M)_0(f) = 0$ for all $f \in \Sigma$. For the induction step $k+1$ we assume for some fixed $k \in \mathbb{N}$ the induction hypothesis

$$R(L)_k(f) \leq R(M)_k(f) \tag{4.19}$$

for all $f \in \Sigma$. In order to show $R(L)_{k+1}(f) \leq R(M)_{k+1}(f)$ we apply Definition 4.45 and show

$$\begin{aligned} &\sup_{r \in R_f} \max_{i=1}^{\mathrm{r}(f)} \left(\lambda \cdot L_k(\mathrm{trgt}(r), x_i) + \sum_{\mu \in \mathrm{der}(r, x_i)} L_k(\mathrm{trgt}(r), \mu) \right) \leq \\ &\sup_{r \in R_f} \max_{i=1}^{\mathrm{r}(f)} \left(\lambda \cdot M_k(\mathrm{trgt}(r), x_i) + \sum_{\mu \in \mathrm{der}(r, x_i)} M_k(\mathrm{trgt}(r), \mu) \right) \end{aligned} \tag{4.20}$$

for each $f \in \Sigma$. First, note that the lifting of Lipschitz factor assignments L and M preserves the order of the induction hypothesis (Equation 4.19), i.e.

$$L_k(\theta, \zeta) \leq M_k(\theta, \zeta)$$

for all $\theta \in \mathsf{DT}(\Sigma)$ and each $\zeta \in \mathscr{V}$. By monotonicity of summation, multiplication with positive scalar, max and sup (i.e. the operators used to define $R(L)_{k+1}$ and $R(M)_{k+1}$, Definition 4.45) Equation 4.20 is given and we derive now

$$R(L)_{k+1}(f) \leq R(M)_{k+1}(f)$$

for each $f \in \Sigma$. To summarize, $R(L)_k(f) \leq R(M)_k(f)$ for each $f \in \Sigma$ and all $k \in \mathbb{N}$. Hence, the proof goal $R(L) \sqsubseteq R(M)$ of Equation 4.18 is given. Thus, R is order-preserving on $(\mathscr{L}_\Sigma, \sqsubseteq)$. □

Definition 4.49 (Canonical LFA)**.** Let $P = (\Sigma, A, R)$ be a PTSS. We call $L_P = \lim_{n \to \infty} R^n(0)$ the *canonical LFA of P*.

Dual to the notion of semantic consistency of LFAs (Definition 4.42) we introduce now the notion of syntactic consistency of LFAs. Intuitively, a syntactically consistent LFA ensures that the Lipschitz factors are compatible with the rules.

Definition 4.50 (Syntactic consistency)**.** Let $P = (\Sigma, A, R)$ be a PTSS and $L \in \mathscr{L}_\Sigma$ some LFA. We call L *consistent with* P (or alternatively L is P-consistent) if $R(L) \sqsubseteq L$.

In other words, all prefixed points of R are consistent with P. In particular, the canonical LFA L_P is consistent with P. Moreover, L_P is the least LFA consistent with P. The syntactic consistency condition $R(L) \sqsubseteq L$ of LFA L with a specification $P = (\Sigma, A, R)$ is a syntactical invariance condition on P that mimics the semantical bisimulation invariance condition $\mathbf{B}(\mathbf{d}) \sqsubseteq \mathbf{d}$ on the induced model $(\mathsf{T}(\Sigma), A, \longrightarrow)$.

Semantic consistency of a LFA L (Definition 4.42) means consistency of L with the bisimilarity metric $\mathbf{d}$ on the induced model $(\mathsf{T}(\Sigma), A, \longrightarrow)$, whereas syntactic consistency of L (Definition 4.50) means consistency of L with the specification $P = (\Sigma, A, R)$ from which the model is derived. As expected, syntactic consistency implies semantic consistency.

Proposition 4.51 (Syntactic consistency implies semantic consistency)**.** *Let $P = (\Sigma, A, R)$ be a PTSS and $L \in \mathscr{L}_\Sigma$ a LFA. If L is consistent with P then L is also consistent with $\mathbf{d}$.*

Proof. By Definition 4.42, L is consistent with $\mathbf{d}$ iff L_k is consistent with $\mathbf{d}_k$ for all $k \in \mathbb{N}$. Hence we have to prove that for an arbitrary $k \in \mathbb{N}$, it holds that L is consistent with $\mathbf{d}_k$. We proceed by induction over k. The base case $k = 0$ is immediate since $\mathbf{d}_0 = \mathbf{0}$ is the constant zero function and any $L \in \mathscr{L}_\Sigma$ is consistent with $\mathbf{d}_0$. The inductive step $k + 1$ follows directly by Proposition 4.46, relation $R(L) \sqsubseteq L$, and the observation that consistency is preserved by the order relation $\sqsubseteq$ on $\mathscr{L}_\Sigma$, i.e. for any $M, M' \in \mathscr{L}_\Sigma$ with $M \sqsubseteq M'$, if M is consistent with $\mathbf{d}_{k+1}$, then M' is consistent with $\mathbf{d}_{k+1}$. □

4.6.2 Uniformly continuous operators

A P-consistent LFA allows us to derive for each operator f an upper bound on the distance between f-composed terms.

Definition 4.52 (Induced upper bound)**.** Let $P = (\Sigma, A, R)$ be a PTSS and $L \in \mathscr{L}_\Sigma$ a LFA. We define for any n-ary operator $f \in \Sigma$ the *upper bound on the distance of f-composed processes induced by L* as the mapping $\omega_{L,f} : (\mathbb{R}_{\geq 0})^n \to \mathbb{R}^\infty_{\geq 0}$ defined by

$$\omega_{L,f}(\epsilon_1, \ldots, \epsilon_n) = \inf_{k \in \mathbb{N}} \left(L_k(f) \sum_{i=1}^{n} \epsilon_i + \lambda^k \right)$$

If L is consistent with P, then $\omega_{L,f}$ is an upper bound on the distance between f-composed terms w.r.t. $\mathbf{d}$.

Theorem 4.53. *Let $P = (\Sigma, A, R)$ be a PTSS and $L \in \mathscr{L}_\Sigma$ a LFA consistent with P. Then*

$$\mathbf{d}(f(s_1, \ldots, s_n), f(t_1, \ldots, t_n)) \leq \omega_{L,f}(\mathbf{d}(s_1, t_1), \ldots, \mathbf{d}(s_n, t_n))$$

for all operators $f \in \Sigma$ and terms $s_i, t_i \in \mathsf{T}(\Sigma)$.

Proof. Remember that by Proposition 4.51 the syntactic consistency of L implies semantic consistency of L, namely we have that L is consistent with $\mathbf{d}$, which means that L is consistent with $\mathbf{d}_k$ for all $k \in \mathbb{N}$. Then we have

$$\mathbf{d}(f(s_1, \ldots, s_n), f(t_1, \ldots, t_n))$$

$$
\begin{aligned}
&\leq \inf_{k\in\mathbb{N}} \left(\mathbf{d}_k(f(s_1,\ldots,s_n),f(t_1,\ldots,t_n)) + \lambda^k\right) && \text{(by Proposition 4.32)}\\
&\leq \inf_{k\in\mathbb{N}} \left(L_k(f)\sum_{i=1}^{n}\mathbf{d}_k(s_i,t_i) + \lambda^k\right) && \text{(since } L \text{ is consistent with } \mathbf{d}_k)\\
&\leq \inf_{k\in\mathbb{N}} \left(L_k(f)\sum_{i=1}^{n}\mathbf{d}(s_i,t_i) + \lambda^k\right) && \text{(by } \mathbf{d}_k \sqsubseteq \mathbf{d})\\
&= \omega_{L,f}(\mathbf{d}(s_1,t_1),\ldots,\mathbf{d}(s_n,t_n)).
\end{aligned}
$$

□

Moreover, if L is consistent with P, then $\omega_{L,f}$ is a modulus of continuity of f w.r.t. $\mathbf{d}$ if all Lipschitz factors $L_k(f)$ of f are finite (c.f. Theorem 4.33).

Theorem 4.54. *Let $P = (\Sigma, A, R)$ be a PTSS and $L \in \mathscr{L}_\Sigma$ a LFA consistent with P. An operator $f \in \Sigma$ is*

1. *uniformly continuous if $L_k(f) < \infty$ for all $k \in \mathbb{N}$,*
2. *Lipschitz continuous if $\sup_{k\in\mathbb{N}} L_k(f) < \infty$, and*
3. *K-Lipschitz continuous if $L_k(f) \leq K$ for all $k \in \mathbb{N}$.*

Proof. We start with Theorem 4.54.1. Since L is consistent with P, by Proposition 4.51 we get that L is consistent with $\mathbf{d}$, namely L_k is consistent with $\mathbf{d}_k$ for all $k \in \mathbb{N}$. Formally, for all $k \in \mathbb{N}$ we have (Definition 4.42):

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

for all $s_i, t_i \in \mathsf{T}(\Sigma)$. By the assumption $L_k(f) < \infty$, we infer that f is $L_k(f)$-Lipschitz continuous w.r.t. $\mathbf{d}_k$. Hence, f is Lipschitz continuous w.r.t. $\mathbf{d}_k$ for each $k \in \mathbb{N}$. By Theorem 4.33 we get that f is uniformly continuous w.r.t. $\mathbf{d}$.

We proceed with Theorem 4.54.2. Since, $\sup_{k\in\mathbb{N}} L_k(f) < \infty$, let $K = \sup_{k\in\mathbb{N}} L_k(f)$. Then by Theorem 4.54.3 we get that f is K-Lipschitz continuous. Hence, f is Lipschitz continuous.

Finally, we show Theorem 4.54.3. By Theorem 4.53 we know that

$$\mathbf{d}(f(s_1,\ldots,s_n),f(t_1,\ldots,t_n)) \leq \inf_{k\in\mathbb{N}}\left(L_k(f)\sum_{i=1}^{n}\mathbf{d}(s_i,t_i) + \lambda^k\right)$$

for all terms $s_i, t_i \in \mathsf{T}(\Sigma)$. Remind that $\lambda < 1$. Since $L_k(f) \leq K$ for all $k \in \mathbb{N}$, we get

$$
\begin{aligned}
&\inf_{k\in\mathbb{N}}\left(L_k(f)\sum_{i=1}^{n}\mathbf{d}(s_i,t_i) + \lambda^k\right)\\
&\leq \inf_{k\in\mathbb{N}}\left(K\sum_{i=1}^{n}\mathbf{d}(s_i,t_i) + \lambda^k\right) && \text{(since } L_k(f) \leq K)
\end{aligned}
$$

$$= K \sum_{i=1}^{n} \mathbf{d}(s_i, t_i) \qquad (\text{since } \inf_{k \in \mathbb{N}} \lambda^k = 0)$$

Hence, f is K-Lipschitz continuous (Definition 4.31.2). □

Hence, if f is Lipschitz continuous, then $\sup_{k\in\mathbb{N}} L_k(f)$ is a Lipschitz factor of f. Since the canonical LFA L_P is the least LFA consistent with P it suffices to verify the conditions of Theorem 4.54 on the canonical LFA.

We provide now an example that shows how to derive the canonical LFA, how to compute the modulus of continuity, and how to determine the resp. compositionality property.

Example 4.55. Let $P = (\Sigma, A, R)$ be the PTSS specifying the synchronous parallel composition operator (Example 4.14) and the copy operator (Example 4.34). Let $L \in \mathscr{L}_\Sigma$ be defined as $L_0(|) = 0$ and $L_0(\mathsf{cp}) = 0$, and $L_k(|) = 1$ and $L_k(\mathsf{cp}) = 2^k$ for any $k \in \mathbb{N}_{>0}$. First we show that L is the canonical LFA $L_P = \lim_{n\to\infty} R^n(0)$ (Definition 4.49). Observe that $R^{n+1}(0)_k = R^n(0)_k$ for all $k \leq n$. Hence, we need to prove that $R^n(0)_n = L_n$ for each $n \in \mathbb{N}$. We reason by induction over n. The base case $R^0(0)_0(|) = 0 = L_0(|)$ and $R^0(0)_0(\mathsf{cp}) = 0 = L_0(\mathsf{cp})$ is immediate. The induction step is $R^{n+1}(0)_{n+1}(|) = \max(\lambda \cdot R^n(0)_n(\mu \mid \nu, x) + R^n(0)_n(\mu \mid \nu, \mu), \lambda \cdot R^n(0)_n(\mu \mid \nu, y) + R^n(0)_n(\mu \mid \nu, \nu)) =$ (inductive hypothesis) $\max(\lambda \cdot L_n(\mu \mid \nu, x) + L_n(\mu \mid \nu, \mu), \lambda \cdot L_n(\mu \mid \nu, y) + L_n(\mu \mid \nu, \nu)) = \max(0 + 1, 0 + 1) = 1 = L_{n+1}(|)$, and $R^{n+1}(0)_{n+1}(\mathsf{cp}) = \max(R^n(0)_n(\mu, \mu), R^n(0)_n(\mathsf{cp}(\mu) \mid \mathsf{cp}(\nu), \mu) + R^n(0)_n(\mathsf{cp}(\mu) \mid \mathsf{cp}(\nu), \nu)) =$ (inductive hypothesis) $\max(L_n(\mu, \mu), L_n(\mathsf{cp}(\mu) \mid \mathsf{cp}(\nu), \mu) + L_n(\mathsf{cp}(\mu) \mid \mathsf{cp}(\nu), \nu)) = \max(1, 2^n + 2^n) = 2^{n+1} = L_{n+1}(\mathsf{cp})$. Hence, we can conclude that L is the canonical LFA. Then, by Theorem 4.53 we get that $\omega_{(L,|)}(\epsilon_1, \epsilon_2) = \epsilon_1 + \epsilon_2$ and $\omega_{(L,\mathsf{cp})}(\epsilon) = \inf_{k\in\mathbb{N}}(2^k\epsilon + \lambda^k)$ are upper bounds for $|$ and cp w.r.t. $\mathbf{d}$. By Theorem 4.54 get that the operator $|$ is 1-Lipschitz continuous and that the operator cp is uniformly continuous. Moreover, the upper bounds are indeed moduli of continuity.

4.6.3 From modulus of continuity to operator specifications

In reverse, we derive now from any modulus of continuity ω a LFA L s.t. any PTSS P consistent with L specifies an operator that has ω as modulus of continuity. The derived LFA depends on ω and the underlying model of process replication. The model of process replication is given as a mapping $\chi : \mathbb{R}_{\geq 0} \times \mathbb{N} \to \mathbb{R}_{\geq 0}$ assigning to each step k an upper bound on the number of spawned process instances. The first argument is a fixed growth factor.

Definition 4.56 (Growth function)**.** We define the following *growth functions* $\chi : \mathbb{R}_{\geq 0} \times \mathbb{N} \to \mathbb{R}_{\geq 0}$

1. $\chi(c, k) = c$ (constant),
2. $\chi(c, k) = c \cdot k$ (linear growth),
3. $\chi(c, k) = c^k$ (exponential growth).

The constant growth function expresses that at most c process instances are spawned irrespective of for how many steps the combined process evolves (cf. non-recurring process replication, Examples 4.14–4.16). The linear growth function will be used to model operators with bounded stepwise replication (cf. recurring step-bounded process replication, Example 4.24 and Corollary 4.25). Similarly, the exponential growth function allows us to model continuously replicating operators (cf. recurring step-unbounded process replication, Example 4.34).

Definition 4.57 (LFA induced by ω and χ)**.** Assume a function $\omega\colon [0,1]^n \to [0,1]$ such that $\omega(0,\ldots,0) = 0$ and $\lim_{(\epsilon_1,\ldots,\epsilon_n)\to(0,\ldots,0)} \omega(\epsilon_1,\ldots,\epsilon_n) = \omega(0,\ldots,0)$, a growth function χ and an operator $f \in \Sigma$. The *LFA* $L^f_{\omega,\chi}$ *induced by* ω *and* χ *for* f is defined by

$$L^f_{\omega,\chi,k}(g) = \begin{cases} \chi(C,k) & \text{if } g = f \\ \infty & \text{if } g \neq f \end{cases}$$

with $C = \sup\left\{c \in \mathbb{R}_{\geq 0} \mid \forall L \in \mathscr{L}_\Sigma.\left((\forall k \in \mathbb{N}.L_k(f) = \chi(c,k)) \Rightarrow \omega_{L,f} \leq \omega\right)\right\}$.

The LFA induced by the exponential growth function is the LFA arising from maximal recurring process replications. The recurring process replication factor C is the maximal process replication per single transition step (possibly repeated along the evolution of the combined process).

Theorem 4.58. *Let $P = (\Sigma, A, R)$ be a PTSS and $L^f_{\omega,\chi}$ be the LFA induced by ω and χ for an operator $f \in \Sigma$. If there exists a P-consistent LFA $L \in \mathscr{L}_\Sigma$ with $L \sqsubseteq L^f_{\omega,\chi}$, then P specifies f s.t. f admits ω as modulus of continuity.*

Proof. Since L is P-consistent, by Theorem 4.53 we get that $\omega_{L,f}$ is an upper bound on the distance between f-composed processes. We get $\omega_{L,f} \leq \omega$ by

$$\begin{aligned}
&\omega_{L,f}(\epsilon_1,\ldots,\epsilon_n) \\
&= \inf_{k\in\mathbb{N}}\left(L_k(f)\sum_{i=1}^{n}\epsilon_i + \lambda^k\right) && \text{(Definition 4.52)} \\
&\leq \inf_{k\in\mathbb{N}}\left(L^f_{\omega,\chi,k}(f)\sum_{i=1}^{n}\epsilon_i + \lambda^k\right) && (L \sqsubseteq L^f_{\omega,\chi}) \\
&= \inf_{k\in\mathbb{N}}\left(\chi(C,k)\sum_{i=1}^{n}\epsilon_i + \lambda^k\right) && \text{(Definition of } L^f_{\omega,\chi,k}) \\
&\leq \omega(\epsilon_1,\ldots,\epsilon_n) && \text{(Definition of } C)
\end{aligned}$$

thus implying that ω is an upper bound on the distance between f-composed processes. Since by definition $\omega(0,\ldots,0) = 0$ and ω is continuous at 0, we get that ω is a modulus of continuity of f. □

Example 4.59. To define a unary operator that may not increase the behavioral distance of its argument, assume the modulus of continuity $\omega(\epsilon) = \epsilon$ (1-Lipschitz continuity). The LFA $L^f_{\omega,\chi}$ induced by ω and $\chi(1,k) = 1$ for f (Definition 4.56.1, Definition 4.57,

Definition 4.52) gives $L^f_{\omega,\chi,k}(f) = 1$. Let f be the operator specified by the rule (with $\theta \in \mathbb{DT}(\Sigma)$ be any distribution term)

$$\frac{x \xrightarrow{a} \mu}{f(x) \xrightarrow{a} \theta}$$

Clearly, $\theta = \mu$ specifies operator f s.t. $L^f_{\omega,\chi}$ is consistent with P (Definition 4.50) and that operator f admits as modulus of continuity ω (Theorem 4.58). Let $t \in \mathsf{T}(\Sigma)$ be any closed term describing some alternative process behavior. With the same argument, also $\theta = \delta(\mu \mid \mu) \oplus_p \delta(t)$ with $p \leq 1/2$ (2 instances proceed with probability at most $1/2$), $\theta = \delta(x \mid x) \oplus_p \delta(t)$ with $p \leq 1/(2\lambda)$ (2 instances proceed with one step delay with probability at most $1/(2\lambda)$), and $\theta = \delta(a^n.(x \mid x)) \oplus_p \delta(t)$ with $p \leq 1/(2\lambda^{n+1})$ ($a^n.\_$ is action prefix operator performing n-times action a followed by the argument process) specify each operator f admitting ω as modulus of continuity (Theorem 4.58).

We conclude by observing that $\theta = f(\mu) \mid \mu$ specifies operator f s.t. P is consistent with the LFA $L^f_{\omega,\chi}$ whereby $L^f_{\omega,\chi,k}(f) = 2k$ is obtained from the linear growth function $\chi(2,k) = 2k$ and the modulus of continuity $\omega(\epsilon) = \inf_{k\in\mathbb{N}}(2k\epsilon + \lambda^k)$. In the same way we can derive that $\theta = f(\mu) \mid f(\mu)$ specifies operator f s.t. P is consistent with the LFA $L^f_{\omega,\chi}$ whereby $L^f_{\omega,\chi,k}(f) = 2^k$ is obtained from the exponential growth function $\chi(2,k) = 2^k$ and the modulus of continuity $\omega(\epsilon) = \inf_{k\in\mathbb{N}}(2^k\epsilon + \lambda^k)$.

4.6.4 Syntactic and semantic compositionality

LFAs induced by moduli of continuity and growth functions (Definition 4.57) are compositional. This allows us to determine the LFA for multiple operators separately, and then to specify those operators simultaneously in a specification consistent with the composed LFAs.

Theorem 4.60. *Let $P = (\Sigma, A, R)$ be a PTSS and $G \subseteq \Sigma$ be a set of operators. For each $g \in G$ let $L^g_{\omega_g,\chi_g}$ be the LFA induced by some ω_g and χ_g for g. If for each $g \in G$ the LFA $L^g_{\omega_g,\chi_g}$ is consistent with P, then also the LFA $\inf_{g\in G} L^g_{\omega_g,\chi_g}$ is consistent with P.*

Proof. Since each $L^g_{\omega_g,\chi_g}$ with $g \in G$ is consistent with P, we have $R(L^g_{\omega_g,\chi_g}) \sqsubseteq L^g_{\omega_g,\chi_g}$ for each $g \in G$ (Definition 4.50). It follows $\inf_{g\in G} R(L^g_{\omega_g,\chi_g}) \sqsubseteq \inf_{g\in G} L^g_{\omega_g,\chi_g}$ (Proposition 4.41). By the monotonicity of R (Proposition 4.48) we have $R(\inf_{g\in G} L^g_{\omega_g,\chi_g}) \sqsubseteq \inf_{g\in G} R(L^g_{\omega_g,\chi_g})$. Thus, $R(\inf_{g\in G} L^g_{\omega_g,\chi_g}) \sqsubseteq \inf_{g\in G} L^g_{\omega_g,\chi_g}$, namely $\inf_{g\in G} L^g_{\omega_g,\chi_g}$ is consistent with P (Definition 4.50). □

Upper bounds of operators (Definition 4.30) are compositional. Hence, we define now an upper bound on the distance between two closed instances of a term by composing the moduli of continuity of the operators of that term. In essence, the following theorem lifts Theorem 4.53 to terms.

Theorem 4.61. *Let $P = (\Sigma, A, R)$ be a PTSS, $L \in \mathscr{L}_\Sigma$ a LFA consistent with P and $t \in \mathbb{T}(\Sigma)$ any open term. For all closed substitutions $\sigma_1, \sigma_2 \colon \mathcal{V} \to T(\Sigma)$ we get*

$$\mathbf{d}(\sigma_1(t), \sigma_2(t)) \leq \inf_{k \in \mathbb{N}} \left(\sum_{x \in \mathcal{V}_s} L_k(t, x) \cdot \mathbf{d}(\sigma_1(x), \sigma_2(x)) + \lambda^k \right)$$

Proof.

$$\begin{aligned}
&\mathbf{d}(\sigma_1(t), \sigma_2(t)) \\
&\leq \inf_{k \in \mathbb{N}} \left(\mathbf{d}_k(\sigma_1(t), \sigma_2(t)) + \lambda^k \right) && \text{(Proposition 4.32)} \\
&\leq \inf_{k \in \mathbb{N}} \left(\sum_{x \in \mathcal{V}_s} L_k(t, x) \cdot \mathbf{d}_k(\sigma_1(x), \sigma_2(x)) + \lambda^k \right) && \text{(Proposition 4.44)} \\
&\leq \inf_{k \in \mathbb{N}} \left(\sum_{x \in \mathcal{V}_s} L_k(t, x) \cdot \mathbf{d}(\sigma_1(x), \sigma_2(x)) + \lambda^k \right) && (\mathbf{d}_k \sqsubseteq \mathbf{d}).
\end{aligned}$$

□

Example 4.62. We start by exemplifying Theorem 4.60. We consider the specification $P = (\Sigma, A, R)$ of operators $G = \{\_ \mid \_, \mathsf{cp}(\_)\}$. As we showed earlier in Example 4.55 the LFAs $L^{|}_{\omega_|, \chi_|, k}(|) = 1$, $L^{|}_{\omega_|, \chi_|, k}(\mathsf{cp}) = \infty$ and $L^{\mathsf{cp}}_{\omega_{\mathsf{cp}}, \chi_{\mathsf{cp}}, k}(\mathsf{cp}) = 2^k$, $L^{\mathsf{cp}}_{\omega_{\mathsf{cp}}, \chi_{\mathsf{cp}}, k}(|) = \infty$ (Definition 4.57) are consistent with P. Then by Theorem 4.60 $L = \inf_{g \in G} L^g_{\omega_g, \chi_g}$ with $L_k(|) = 1$ and $L_k(\mathsf{cp}) = 2^k$ is consistent with P.

We proceed by exemplifying Theorem 4.61. Consider terms $t = \mathsf{cp}(x \mid x)$. By using $L_k(|) = 1$ and $L_k(\mathsf{cp}) = 2^k$ (Example 4.55), we get (Definition 4.43) $L_k(\mathsf{cp}(x \mid x), x) = L_k(\mathsf{cp}) \cdot L_k(x \mid x, x) = 2^k \cdot (L_k(|) \cdot (L_k(x, x) + L_k(x, x)) = 2^{k+1}$. Hence, by Theorem 4.61 we get $\mathbf{d}(\sigma_1(t), \sigma_2(t)) \leq \inf_{k \in \mathbb{N}}(2^{k+1} \cdot \mathbf{d}(\sigma_1(x), \sigma_2(x)) + \lambda^k)$ for all closed substitutions $\sigma_1, \sigma_2 \colon \mathcal{V} \to \mathsf{T}(\Sigma)$. Equally, for $t = \mathsf{cp}(x) \mid \mathsf{cp}(x)$ we get $L_k(\mathsf{cp}(x) \mid \mathsf{cp}(x), x) = 2^{k+1}$ and $\mathbf{d}(\sigma_1(t), \sigma_2(t)) \leq \inf_{k \in \mathbb{N}}(2^{k+1} \cdot \mathbf{d}(\sigma_1(x), \sigma_2(x)) + \lambda^k)$. The nesting of the copy operator $\mathsf{cp}(\mathsf{cp}(x))$ induces $L_k(\mathsf{cp}(\mathsf{cp}(x)), x) = 2^{2k}$ with distance bound $\mathbf{d}(\sigma_1(\mathsf{cp}(\mathsf{cp}(x))), \sigma_2(\mathsf{cp}(\mathsf{cp}(x)))) \leq \inf_{k \in \mathbb{N}}(2^{2k} \cdot \mathbf{d}(\sigma_1(x), \sigma_2(x)) + \lambda^k)$.

4.7 Deciding the compositionality property

The rule formats developed in the former sections determine simultaneously the compositionality property of all specified operators. To exemplify this, consider a specification $P_1 = (\Sigma_1, A, R_1)$ and a conservative extension $P_2 = (\Sigma_2, A, R_2) \sqsupseteq P_1$. Given a Lipschitz factor assignment $L_1 \in \mathscr{L}_{\Sigma_1}$ consistent with P_1, then for any Lipschitz factor assignment $L_2 \in \mathscr{L}_{\Sigma_2}$ consistent with P_2, the Lipschitz factors for the operators in $\Sigma_2 \setminus \Sigma_1$ may depend on the Lipschitz factors L_1 for operators in Σ_1. For instance, considering Example 4.14 and PTSSs P_1 specifying operator $\_ \mid \_$ and P_2 specifying operator f. Let r_f be the rule specifying operator f. If $\theta = \mathrm{trgt}(r_f)$ contains operator $\_ \mid \_$, then the Lipschitz factor $L(f)$ of operator $f \in \Sigma_2 \setminus \Sigma_1$ depends on the Lipschitz factor $L(|)$ of operator $| \in \Sigma_1$. In other words, (except if P_2 is disjoint to P_1), in order to decide which compositionality

property the newly specified operators in $\Sigma_2 \setminus \Sigma_1$ have we need to consider also the compositionality properties of the operators in Σ_1. Thus, the compositonality property of an operator in the extension $f \in \Sigma_2 \setminus \Sigma_1$ cannot (in general) be decided by only inspecting the specification rules $r \in R_f \subseteq R_2 \setminus R_1$.

We investigate now which compositionality properties allow for a local decision procedure, i.e. for which the compositionality property of an operator can be decided by only inspecting the specification rules of that operator with mild assumptions on the remaining operators.

Definition 4.63 (Finitely branching and finitely copying PTSS)**.** Let $P = (\Sigma, A, R)$ be any PTSS. We call P

- *finitely branching* if for each operator $f \in \Sigma$ the set of rules R_f specifying f is finite,
- *finitely copying* if there are mappings $K_d, K_v \colon \Sigma \to \mathbb{N}$ s.t. for each $f \in \Sigma$ we have
  - $\mathsf{depth}(\mathsf{trgt}(r)) \leq K_d(f)$, and
  - $|\mathsf{Var}(\mathsf{trgt}(r))| \leq K_v(f)$

  for all rules $r \in R_f$.

Intuitively, $K_d(f)$ (resp. $K_v(f)$) gives an upper bound on the depth (resp. an upper bound on the number of variables) of the targets of rules specifying operator f. We remark that the property of finitely branching PTSS bases on the cardinality of the set of specification rules, while the property of finitely copying PTSS bases on the syntactical structure of the specification rules. It is not hard to show that if a PTSS P is finitely branching then P is finitely copying. We will show that each finitely copying PTSS specifies uniformly continuous operators (Theorem 4.66).

Lemma 4.64. *If a PTSS $P = (\Sigma, A, R)$ is finitely branching, then P is finitely copying.*

Proof. We construct the mappings $K_d, K_v \colon \Sigma \to \mathbb{N}$ witnessing that P is finitely copying as follows:

$$K_d(f) = \max_{r \in R_f} \mathsf{depth}(\mathsf{trgt}(r))$$
$$K_v(f) = \max_{r \in R_f} |\mathsf{Var}(\mathsf{trgt}(r))|$$

for all operators $f \in \Sigma$. Note that K_d and K_v are well-defined since R_f, $\mathsf{depth}(\mathsf{trgt}(r))$ and $|\mathsf{Var}(\mathsf{trgt}(r))|$ are all finite. It is immediate that for each operator $f \in \Sigma$ we have

$$\mathsf{depth}(\mathsf{trgt}(r)) \leq K_d(f)$$
$$|\mathsf{Var}(\mathsf{trgt}(r))| \leq K_v(f)$$

for all rules $r \in R_f$. □

Notation 4.65. For state and distribution terms we define by $\mathrm{ops} \colon \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma) \to P(F)$ the mapping that gives for $\mathrm{ops}(t)$ the set of all operator symbols in state term $t \in \mathbb{T}(\Sigma)$, defined as $\mathrm{ops}(x) = \emptyset$ and $\mathrm{ops}(f(t_1, \ldots, t_n)) = \{f\} \cup \bigcup_{i=1}^{n} \mathrm{ops}(t_i)$, and for $\mathrm{ops}(\theta)$ the set of all operator symbols in distribution term $\theta \in \mathbb{DT}(\Sigma)$, defined as $\mathrm{ops}(\mu) = \emptyset$, $\mathrm{ops}(\delta(t)) = \mathrm{ops}(t)$, $\mathrm{ops}(\sum_{i \in I} p_i \theta_i) = \bigcup_{i \in I} \mathrm{ops}(\theta_i)$ and $\mathrm{ops}(f(\theta_1, \ldots, \theta_n)) = \{f\} \cup \bigcup_{i=1}^{n} \mathrm{ops}(\theta_i)$.

Theorem 4.66. *Let $P = (\Sigma, A, R)$ be a PTSS.*

1. *If P is finitely branching, then each $f \in \Sigma$ is uniformly continuous w.r.t.* $\mathbf{d}$.
2. *If P is finitely copying, then each $f \in \Sigma$ is uniformly continuous w.r.t.* $\mathbf{d}$.

Before proving Theorem 4.66 we develop first an upper bound on the Lipschitz factor of terms from the Lipschitz factors of the operators and the term depth.

Lemma 4.67. *Let $P = (\Sigma, A, R)$ be a PTSS and $L \in \mathscr{L}_\Sigma$ any Lipschitz factor assignment. Then*

$$L_k(t,\zeta) \leq \begin{cases} 1 & \text{if } ops(t) = \emptyset \\ |Var(t)| \left(\max_{g \in ops(t)} \overline{L_k(g)} \right)^{depth(t)} & \text{if } ops(t) \neq \emptyset \end{cases}$$

for all $k \in \mathbb{N}$, state or distribution term $t \in \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)$ and variable $\zeta \in \mathcal{V}$.

Proof. Consider arbitrary $k \in \mathbb{N}$ and $\zeta \in \mathcal{V}$. Then we start with state terms $t \in \mathbb{T}(\Sigma)$. We reason by induction over the structure of t. The base case $t = x \in \mathcal{V}_s$ is immediate since by definition $L_k(x,\zeta) \in \{0,1\}$, hence $L_k(x,\zeta) \leq 1$ (case $\mathrm{ops}(x) = \emptyset$). We proceed with the inductive step $t = f(t_1, \ldots, t_n)$ and show $L_k(t,\zeta) \leq |\mathsf{Var}(t)| \left(\max_{g \in \mathrm{ops}(t)} \overline{L_k(g)} \right)^{\mathsf{depth}(t)}$ (since $\mathrm{ops}(t) \neq \emptyset$). Then

$$\begin{aligned}
& L_k(t,\zeta) \\
=& L_k(f) \sum_{i=1}^{n} L_k(t_i,\zeta) \\
\leq& L_k(f) \sum_{i=1}^{n} \begin{cases} 1 & \text{if } \mathrm{ops}(t_i) = \emptyset \\ |\mathsf{Var}(t_i)| \left(\max_{g \in \mathrm{ops}(t_i)} \overline{L_k(g)} \right)^{\mathsf{depth}(t_i)} & \text{if } \mathrm{ops}(t_i) \neq \emptyset \end{cases} \\
\leq& L_k(f) \sum_{i=1}^{n} \begin{cases} 1 & \text{if } \mathrm{ops}(t_i) = \emptyset \\ |\mathsf{Var}(t_i)| \left(\max_{g \in \mathrm{ops}(t)} \overline{L_k(g)} \right)^{\mathsf{depth}(t_i)} & \text{if } \mathrm{ops}(t_i) \neq \emptyset \end{cases} \\
\leq& L_k(f) \sum_{i=1}^{n} |\mathsf{Var}(t_i)| \left(\max_{g \in \mathrm{ops}(t)} \overline{L_k(g)} \right)^{\mathsf{depth}(t_i)} \\
\leq& \left(\max_{g \in \mathrm{ops}(t)} \overline{L_k(g)} \right) \sum_{i=1}^{n} |\mathsf{Var}(t_i)| \left(\max_{g \in \mathrm{ops}(t)} \overline{L_k(g)} \right)^{\mathsf{depth}(t_i)} \\
\leq& \left(\max_{g \in \mathrm{ops}(t)} \overline{L_k(g)} \right) \sum_{i=1}^{n} |\mathsf{Var}(t_i)| \left(\max_{g \in \mathrm{ops}(t)} \overline{L_k(g)} \right)^{\max_{i=1}^{n} \mathsf{depth}(t_i)} \\
=& \left(\sum_{i=1}^{n} |\mathsf{Var}(t_i)| \right) \left(\max_{g \in \mathrm{ops}(t)} \overline{L_k(g)} \right) \left(\max_{g \in \mathrm{ops}(t)} \overline{L_k(g)} \right)^{\max_{i=1}^{n} \mathsf{depth}(t_i)} \\
=& |\mathsf{Var}(t)| \left(\max_{g \in \mathrm{ops}(t)} \overline{L_k(g)} \right)^{1+\max_{i=1}^{n} \mathsf{depth}(t_i)} \\
=& |\mathsf{Var}(t)| \left(\max_{g \in \mathrm{ops}(t)} \overline{L_k(g)} \right)^{\mathsf{depth}(t)}
\end{aligned}$$

with step 1 by the definition of L_k, step 2 by the inductive hypothesis, step 3 by $\mathrm{ops}(t_i) \subseteq \mathrm{ops}(t)$, step 4 from the observations that in case $\mathrm{ops}(t_i) = \emptyset$, i.e. $t_i = x \in \mathcal{V}_s$, we have $|\mathsf{Var}(t_i)|(\max_{g\in\mathrm{ops}(t)} \overline{L_k(g)})^{\mathsf{depth}(t_i)} \geq 1$, since $|\mathsf{Var}(x)| = 1$, $\max_{g\in\mathrm{ops}(t)} \overline{L_k(g)} \geq \overline{L_k(f)} \geq 1$ and $\mathsf{depth}(x) = 0$, step 5 by $f \in \mathrm{ops}(t)$ and the last step by the definition of depth.

Consider now $\theta \in \mathbb{DT}(\Sigma)$ an open distribution term. We reason by induction over the structure of θ. The base case with θ a distribution variable follows exactly as above. The base case with $\theta = \delta(t)$ follows from $L_k(\delta(t), \zeta) = L_k(t, \zeta)$, $\mathsf{Var}(\delta(t)) = \mathsf{Var}(t)$, $\mathsf{depth}(\delta(t)) = \mathsf{depth}(t)$ and the already proved case for $t \in \mathbb{T}(\Sigma)$ an open state term.

Consider now the induction step $\theta = \sum_{i\in I} p_i\theta_i$. We distinguish two cases. The first case is $\mathrm{ops}(\theta) \neq \emptyset$. We have

$$
\begin{aligned}
& L_k(\theta, \zeta) \\
=& \sum_{i=1}^{n} p_i L_k(\theta_i, \zeta) \\
\leq& \sum_{i=1}^{n} p_i \begin{cases} 1 & \text{if } \mathrm{ops}(\theta_i) = \emptyset \\ |\mathsf{Var}(\theta_i)| \left(\max_{g\in\mathrm{ops}(\theta_i)} \overline{L_k(g)}\right)^{\mathsf{depth}(\theta_i)} & \text{if } \mathrm{ops}(\theta_i) \neq \emptyset \end{cases} \\
\leq& \sum_{i=1}^{n} p_i |\mathsf{Var}(\theta)| \left(\max_{g\in\mathrm{ops}(\theta)} \overline{L_k(g)}\right)^{\max_{i=1}^{n} \mathsf{depth}(\theta_i)} \\
=& \sum_{i=1}^{n} p_i |\mathsf{Var}(\theta)| \left(\max_{g\in\mathrm{ops}(\theta)} \overline{L_k(g)}\right)^{\mathsf{depth}(\theta)} \\
=& |\mathsf{Var}(\theta)| \left(\max_{g\in\mathrm{ops}(\theta)} \overline{L_k(g)}\right)^{\mathsf{depth}(\theta)} .
\end{aligned}
$$

with step 1 by the definition of L_k, step 2 by the inductive hypothesis, step 3 following from the observation that $|\mathsf{Var}(\theta)|(\max_{g\in\mathrm{ops}(\theta)} \overline{L_k(g)})^{\max_{i=1}^{n} \mathsf{depth}(\theta_i)} \geq 1$ if $\mathrm{ops}(\theta_i) = \emptyset$, and step 4 from the definition of depth. The second case is $\mathrm{ops}(\theta) = \emptyset$. Then, also $\mathrm{ops}(\theta_i) = \emptyset$ for all $i \in I$ and we get $L_k(\theta, \zeta) = \sum_{i=1}^{n} p_i L_k(\theta_i, \zeta) \leq \sum_{i=1}^{n} p_i = 1$.

Finally, the inductive step $\theta = f(\theta_1, \ldots, \theta_n)$ follows exactly as the case $t = f(t_1, \ldots, t_n)$ above. □

Proof of Theorem 4.66. Since P is finitely copying if P is finitely branching (Lemma 4.64), it suffices to show Theorem 4.66.2. By Theorem 4.54.1 it is enough to provide a Lipschitz factor assignment $L \in \mathscr{L}_\Sigma$ such that L is consistent with P and $L_k(f) < \infty$ for all operators $f \in \Sigma$ and for each $k \in \mathbb{N}$. We will show that the canonical Lipschitz factor assignment L_P (Definition 4.49) satisfies

$$\forall k \in \mathbb{N}.\ \forall f \in \Sigma.\ L_{P,k}(f) < \infty. \tag{4.21}$$

We will show Equation 4.21 by induction over k. Remind that $L_{P,k} = R^k(0)$ by Definition 4.49, with $0 \in \mathscr{L}_\Sigma$. Moreover, note that $R^l(0)_m = R^k(0)_m$ whenever $m \leq l \leq k$. Hence, R^k fixes the first k-Lipschitz factors of all operators.

The base case $k = 0$ is trivial since by definition

$$L_{P,0}(f) = 0 \tag{4.22}$$

for all $f \in \Sigma$. For the induction step assume as induction hypothesis that

$$\forall f \in \Sigma.\ L_{P,k}(f) < \infty \tag{4.23}$$

for any fixed k. Consider any operator $f \in \Sigma$. By Definition 4.45 (with $L_{P,k+1}(f) = R^{k+1}(0)_{k+1}(f)$) we get

$$L_{P,k+1}(f) = \sup_{r \in R_f} \max_{i=1}^{\mathrm{r}(f)} \left(\lambda \cdot L_{P,k}(\mathrm{trgt}(r), x_i) + \sum_{\mu \in \mathrm{der}(r,x_i)} L_{P,k}(\mathrm{trgt}(r), \mu) \right). \tag{4.24}$$

By Lemma 4.67 and the finitely copying assumption we get

$$L_{P,k}(\mathrm{trgt}(r), \zeta) \leq \begin{cases} 1 & \text{if } \mathrm{ops}(\mathrm{trgt}(r)) = \emptyset \\ K_v(f) \left(\max_{g \in \mathrm{ops}(\mathrm{trgt}(r))} \overline{L_{P,k}(g)} \right)^{K_d(f)} & \text{if } \mathrm{ops}(\mathrm{trgt}(r)) \neq \emptyset \end{cases} \tag{4.25}$$

for any $\zeta \in \mathscr{V}$. If $\mathrm{ops}(\mathrm{trgt}(r)) = \emptyset$ then the right hand side of Equation 4.25 is clearly finite. If $\mathrm{ops}(\mathrm{trgt}(r)) \neq \emptyset$ then, since $\mathrm{ops}(\mathrm{trgt}(r))$ is finite, $L_{P,k}(g) < \infty$ by the induction hypothesis Equation 4.23, and $K_d(f)$, $K_v(f)$ are finite by definition, we get that the right hand side of Equation 4.25 is finite. Then

$$\begin{aligned} &\lambda \cdot L_{P,k}(\mathrm{trgt}(r), x_i) + \sum_{\mu \in \mathrm{der}(r,x_i)} L_{P,k}(\mathrm{trgt}(r), \mu) \leq \\ &\qquad (\lambda + K_v(f)) \begin{cases} 1 & \text{if } \mathrm{ops}(\mathrm{trgt}(r)) = \emptyset \\ K_v(f) \left(\max_{g \in \mathrm{ops}(\mathrm{trgt}(r))} \overline{L_{P,k}(g)} \right)^{K_d(f)} & \text{if } \mathrm{ops}(\mathrm{trgt}(r)) \neq \emptyset \end{cases} \end{aligned} \tag{4.26}$$

since $|\mathrm{der}(r, x_i)| \leq K_v(f)$. With the same argumentation as before the right-hand side of Equation 4.26 is finite. Since the right-hand side of Equation 4.26 does not depend on the rule r nor on the index i, and from the arbitrarity of f we derive from Equation 4.24 the inequality

$$\forall f \in \Sigma.\ L_{P,k+1}(f) < \infty \tag{4.27}$$

Hence, by the base case Equation 4.22 and the induction step that Equation 4.23 $\Rightarrow$ Equation 4.27 we derive the proof goal Equation 4.21. Thus, using Theorem 4.54.1 on this Lipschitz factor assignment L_P we conclude that all operators $f \in \Sigma$ are uniformly continuous. □

Since finitely copying PTSSs are closed under finite union, the specification of uniformly continuous operators may be verified compositionally.

Corollary 4.68. *Let P_1 and P_2 be PTSSs. If P_1 and P_2 are finitely copying, then all operators specified by $P_1 \cup P_2$ are uniformly continuous.*

Proof. Assume $P_1 = (\Sigma_1, A, R_1)$ and $P_2 = (\Sigma_2, A, R_2)$. Let $K_d^1, K_v^1 \colon \Sigma_1 \to \mathbb{N}$ be the mappings witnessing that P_1 is finitely copying, and let $K_d^2, K_v^2 \colon \Sigma_2 \to \mathbb{N}$ be the mappings witnessing that P_2 is finitely copying. We define now the mappings $K_d, K_v \colon \Sigma_1 \cup \Sigma_2 \to \mathbb{N}$ as $K_d(f) = \max(K_d^1(f), K_d^2(f))$ and $K_v(f) = \max(K_v^1(f), K_v^2(f))$ for all $f \in \Sigma_1 \cap \Sigma_2$, $K_d(f) = K_d^1(f)$ and $K_v(f) = K_v^1(f)$ for all $f \in \Sigma_1 \setminus \Sigma_2$ and $K_d(f) = K_d^2(f)$ and $K_v(f) = K_v^2(f)$ for all $f \in \Sigma_2 \setminus \Sigma_1$. It is clear that K_d and K_v are mappings witnessing that $P_1 \cup P_2$ is finitely copying. □

4.8 Compositionality w.r.t. any behavioral metric

We developed in the former sections expressive rule and specification formats w.r.t. the bisimilarity metric **d** formed by a bisimulation game (Definition 2.21) using the Kantorovich lifting **K** (Definition 2.17). Now we explore starting from the rule and specification formats which properties of the behavioral metric suffice s.t. the specified operators satisfy the respective compositionality properties. We consider the compositional properties of Lipschitz and uniform continuity.

4.8.1 Lifting functional induced bisimulation metric

We start by generalizing bisimulation metrics and define the notion of bisimulation metric induced by some functional that lifts state metrics to distribution metrics.

Definition 4.69 (Lifting functional induced bisimulation metric)**.** Let $\mathbf{D}\colon [0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)} \to [0,1]^{\Delta(\mathsf{T}(\Sigma))\times\Delta(\mathsf{T}(\Sigma))}$ be any function that lifts 1-bounded pseudometrics on $\mathsf{T}(\Sigma)$ to 1-bounded pseudometrics on $\Delta(\mathsf{T}(\Sigma))$. A 1-bounded pseudometric d on $\mathsf{T}(\Sigma)$ is a **D**-*induced* λ-*bisimulation metric* with $\lambda \in (0,1]$ if for all terms $s,t \in \mathsf{T}(\Sigma)$ with $d(s,t) < 1$, if $s \xrightarrow{a} \pi$ then there exists a transition $t \xrightarrow{a} \pi'$ s.t. $\lambda \cdot \mathbf{D}(d)(\pi,\pi') \leq d(s,t)$.

We will prove that a lifting functional **D** satisfying monotonicity (Proposition 2.32.1) induces a notion of least bisimulation metric (Proposition 4.70). Moreover, if **D** satisfies also non-expansiveness of Dirac-embedding and compatibility with convex combinations (Proposition 2.32.2 and Proposition 2.32.3), and preserves linear moduli of continuity of operators (Corollary 2.34), then the rule and specification formats of the former sections specify operators satisfying the respective compositionality properties w.r.t. the least bisimulation metric induced by **D** (Theorems 4.73, 4.74, 4.76).

For an arbitrary function $\mathbf{D}\colon [0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)} \to [0,1]^{\Delta(\mathsf{T}(\Sigma))\times\Delta(\mathsf{T}(\Sigma))}$, the bisimulation functional $\mathbf{B}_{\mathbf{D},\lambda}\colon [0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)} \to [0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)}$ is defined analogous to Definition 2.23 by

$$\mathbf{B}_{\mathbf{D},\lambda}(d)(s,t) = \sup_{a\in A}\{\mathbf{H}(\lambda\cdot\mathbf{D}(d))(der(s,a),der(t,a))\}$$

for all functions $d\colon \mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma) \to [0,1]$ and terms $s,t \in \mathsf{T}(\Sigma)$.

If **D** is monotone, then $\mathbf{B}_{\mathbf{D},\lambda}$ has the least fixed point on the lattice $([0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)}, \sqsubseteq)$.

Proposition 4.70. *Let* $\mathbf{D}\colon [0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)} \to [0,1]^{\Delta(\mathsf{T}(\Sigma))\times\Delta(\mathsf{T}(\Sigma))}$ *be monotone. Then*

1. $\mathbf{B}_{\mathbf{D},\lambda}$ *is monotone.*
2. $\mathbf{B}_{\mathbf{D},\lambda}$ *has a least fixed point.*

Proof. Lemma 4.70.1 follows directly from the monotonicity of **H** and the fact that the composition of monotone functions yields a monotone function. Lemma 4.70.2 follows from Lemma 4.70.1 and the Knaster-Tarski theorem (since $([0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)}, \sqsubseteq)$ is a complete lattice). □

The prefixed points of $\mathbf{B}_{\mathbf{D},\lambda}$ are precisely the **D**-induced λ-bisimulation metrics.

Lemma 4.71. *Let $d\colon T(\Sigma)\times T(\Sigma)\to[0,1]$ be any pseudometric. Then $\mathbf{B}_{\mathbf{D},\lambda}(d)\sqsubseteq d$ iff d is a $\mathbf{D}$-induced λ-bisimulation metric.*

Proof. We have

$$\begin{aligned}
&\mathbf{B}_{\mathbf{D},\lambda}(d)\sqsubseteq d\\
\iff&\forall s,t\in\mathsf{T}(\Sigma).\sup_{a\in A}\{\mathbf{H}(\lambda\cdot\mathbf{D}(d))(der(s,a),der(t,a))\}\leq d(s,t)\\
\iff&\forall s,t\in\mathsf{T}(\Sigma).\Big(d(s,t)<1\implies\sup_{a\in A}\{\mathbf{H}(\lambda\cdot\mathbf{D}(d))(der(s,a),der(t,a))\}\leq d(s,t)\Big)\\
\iff&\forall s,t\in\mathsf{T}(\Sigma).\big(d(s,t)<1\implies\\
&\qquad\forall a\in A.\forall(s,a,\pi)\in\rightarrow.\exists(t,a,\pi')\in\rightarrow.(\lambda\cdot\mathbf{D}(d))(\pi,\pi')\leq d(s,t)\big)\\
\iff&\ d\text{ is a }\mathbf{D}\text{-induced }\lambda\text{-bisimulation metric.}
\end{aligned}$$

We consider in the second reasoning step only $d(s,t)<1$ because the inequality is trivial if $d(s,t)=1$ since both d and $\mathbf{H}(\lambda\cdot\mathbf{D}(d))$ are 1-bounded pseudometrics. □

Hence, if $\mathbf{D}$ is monotone, then on the one hand there exists a least fixed point of $\mathbf{B}_{\mathbf{D},\lambda}$, on the other hand there exists a smallest $\mathbf{D}$-induced λ-bisimulation metric, and they coincide. We denote by $\mathbf{d}_{\mathbf{D},\lambda}$ the least fixed point of $\mathbf{B}_{\mathbf{D},\lambda}$, and we name it as the $\mathbf{D}$-*induced λ-bisimilarity metric*. Moreover, we denote by $\mathbf{d}_{\mathbf{D},\lambda,k}$ the pseudometric $\mathbf{B}^k_{\mathbf{D},\lambda}(0)$, and we name it as the $\mathbf{D}$-*induced up-to-k λ-bisimilarity metric*.

Proposition 4.72. *The generalized Kantorovich lifting $\mathbf{K}_V$ [Cha+14, Equation 3] with $V=([0,1],d_V)$ built on a linear distance function[4] d_V satisfies Proposition 2.32 and Corollary 2.34.*

Proof. The generalized Kantorovich lifting [Cha+14] is defined by

$$\mathbf{K}_V(d)(\pi,\pi')=\sup\{d_V(\hat{f}(\pi),\hat{f}(\pi'))\mid f\in(\mathsf{T}(\Sigma),d)\to_1(V,d_V)\}$$

with $f\in(\mathsf{T}(\Sigma),d)\to_1(V,d_V)$ iff $f\in(\mathsf{T}(\Sigma),d)\to(V,d_V)$ is a mapping between the metric space of terms $(\mathsf{T}(\Sigma),d)$ and (V,d_V) with $d_V(f(s),f(t))\leq d(s,t)$ for all $s,t\in\mathsf{T}(\Sigma)$, and the lifting $\hat{f}$ is defined by $\hat{f}(\pi)=\sum_{t\in\mathsf{T}(\Sigma)}\pi(t)f(t)$.

Proposition 2.32.1 is proved as Proposition 2 in [Cha+14]. To show Proposition 2.32.2 observe that

$$\begin{aligned}
&d_V(\hat{f}(\delta(s)),\hat{f}(\delta(t)))\\
&=d_V(f(s),f(t))&&\text{(by Definition of }\hat{f}\text{)}\\
&\leq d(s,t)&&\text{(since }f\in(\mathsf{T}(\Sigma),d)\to_1(V,d_V)\text{)}
\end{aligned}$$

for all terms $s,t\in\mathsf{T}(\Sigma)$. Hence, $\mathbf{K}_V(d)(\delta(s),\delta(t))\leq d(s,t)$. To show Proposition 2.32.3 we reason

$$\mathbf{K}_V(d)\left(\sum_{i\in I}p_i\pi_i,\sum_{i\in I}p_i\pi'_i\right)$$

[4] A distance function d_V is linear if $d_V(cx+(1-c)y,cz+(1-c)w)=cd_V(x,z)+(1-c)d_V(y,w)$ for all $x,y,z,w\in[0,1]$ and $c\in[0,1]$.

$$
\begin{aligned}
&= \sup\left\{ d_V\left(\hat{f}\left(\sum_{i\in I} p_i\pi_i\right), \hat{f}\left(\sum_{i\in I} p_i\pi_i'\right)\right) \mid f \in (\mathsf{T}(\Sigma), d) \to_1 (V, d_V)\right\}\\
&= \sup\left\{ d_V\left(\sum_{i\in I} p_i\hat{f}(\pi_i), \sum_{i\in I} p_i\hat{f}(\pi_i')\right) \mid f \in (\mathsf{T}(\Sigma), d) \to_1 (V, d_V)\right\}\\
&= \sup\left\{ \sum_{i\in I} p_i d_V(\hat{f}(\pi_i), \hat{f}(\pi_i')) \mid f \in (\mathsf{T}(\Sigma), d) \to_1 (V, d_V)\right\}\\
&\leq \sum_{i\in I} p_i \sup\left\{ d_V(\hat{f}(\pi_i), \hat{f}(\pi_i')) \mid f \in (\mathsf{T}(\Sigma), d) \to_1 (V, d_V)\right\}\\
&= \sum_{i\in I} p_i\, \mathbf{K}_V(d)(\pi_i, \pi_i')
\end{aligned}
$$

where steps 1 and 5 follow by Definition of $\mathbf{K}_V$, step 2 by Definition of $\hat{f}$ and distributivity of multiplication over addition, and step 3 by linearity of d_V.

To show Corollary 2.34 assume that for some n-ary operator $g \in \Sigma$ and fixed $L \in \mathbb{R}_{\geq 0}$ we have $d(g(s_1,\ldots,s_n), f(t_1,\ldots,t_n)) \leq L \cdot \sum_{i=1}^n d(s_i,t_i)$ for all terms $s_1,t_1,\ldots,s_n,t_n \in \mathsf{T}(\Sigma)$. In the following reasoning we will use in an elementary way that for linear d_V there is some $\alpha \in [0,1]$ s.t.

$$d_V(x,y) = \alpha|x-y|$$

for all $x, y \in [0,1]$. Then we have

$$
\begin{aligned}
&\mathbf{K}_V(d)(g(\pi_1,\ldots,\pi_n), g(\pi_1',\ldots,\pi_n'))\\
&= \sup\left\{ d_V(\hat{f}(g(\pi_1,\ldots,\pi_n)), \hat{f}(g(\pi_1',\ldots,\pi_n'))) \mid f \in (\mathsf{T}(\Sigma), d) \to_1 (V, d_V)\right\}\\
&= \sup\Bigg\{ d_V\Bigg(\sum_{t_1,\ldots,t_n \in \mathsf{T}(\Sigma)} f(g(t_1,\ldots,t_n)) \prod_{i=1}^n \pi_i(t_i), \sum_{t_1',\ldots,t_n' \in \mathsf{T}(\Sigma)} f(g(t_1',\ldots,t_n')) \prod_{i=1}^n \pi_i'(t_i')\Bigg) \mid\\
&\qquad f \in (\mathsf{T}(\Sigma), d) \to_1 (V, d_V)\Bigg\}\\
&= \sup\Bigg\{ d_V\Bigg(\sum_{\substack{t_1,\ldots,t_n\\ t_1',\ldots,t_n' \in \mathsf{T}(\Sigma)}} f(g(t_1,\ldots,t_n)) \prod_{i=1}^n \pi_i(t_i)\pi_i'(t_i'),\\
&\qquad \sum_{\substack{t_1,\ldots,t_n\\ t_1',\ldots,t_n' \in \mathsf{T}(\Sigma)}} f(g(t_1',\ldots,t_n')) \prod_{i=1}^n \pi_i(t_i)\pi_i'(t_i')\Bigg) \mid f \in (\mathsf{T}(\Sigma), d) \to_1 (V, d_V)\Bigg\}\\
&= \sup\Bigg\{ \sum_{\substack{t_1,\ldots,t_n\\ t_1',\ldots,t_n' \in \mathsf{T}(\Sigma)}} \prod_{i=1}^n \pi_i(t_i)\pi_i'(t_i') d_V(f(g(t_1,\ldots,t_n)), f(g(t_1',\ldots,t_n')))\\
&\qquad \mid f \in (\mathsf{T}(\Sigma), d) \to_1 (V, d_V)\Bigg\}\\
&\leq \sum_{\substack{t_1,\ldots,t_n\\ t_1',\ldots,t_n' \in \mathsf{T}(\Sigma)}} \prod_{i=1}^n \pi_i(t_i)\pi_i'(t_i') \alpha d(g(t_1,\ldots,t_n), g(t_1',\ldots,t_n'))
\end{aligned}
$$

$$
\begin{aligned}
&\leq \sum_{\substack{t_1,\dots,t_n\\ t'_1,\dots,t'_n}\in \mathsf{T}(\Sigma)} \prod_{i=1}^{n} \pi_i(t_i)\pi'_i(t'_i) L \sum_{i=1}^{n} \alpha d(t_i,t'_i)\\
&= L \sum_{\substack{t_1,\dots,t_n\\ t'_1,\dots,t'_n}\in \mathsf{T}(\Sigma)} \left(\prod_{i=1}^{n} \pi_i(t_i)\pi'_i(t'_i)\right) \sum_{i=1}^{n} \alpha d(t_i,t'_i)\\
&\leq L \sum_{i=1}^{n} \sum_{t_i,t'_i\in \mathsf{T}(\Sigma)} \pi_i(t_i)\pi'_i(t'_i)\alpha d(t_i,t'_i)\\
&= L \sum_{i=1}^{n} \sum_{t_i,t'_i\in \mathsf{T}(\Sigma)} \pi_i(t_i)\pi'_i(t'_i) \sup\left\{d_V(f(t_i),f(t'_i)) \mid f\in(\mathsf{T}(\Sigma),d)\to_1 (V,d_V)\right\}\\
&= L \sum_{i=1}^{n} \sup\Bigg\{ d_V\Bigg(\sum_{t_i,t'_i\in \mathsf{T}(\Sigma)} f(t_i)\pi_i(t_i)\pi'_i(t'_i), \sum_{t_i,t'_i\in \mathsf{T}(\Sigma)} f(t'_i)\pi_i(t_i)\pi'_i(t'_i)\Bigg) \mid\\
&\qquad\qquad f\in(\mathsf{T}(\Sigma),d)\to_1 (V,d_V)\Bigg\}\\
&= L \sum_{i=1}^{n} \sup\left\{d_V(\hat{f}(\pi_i),\hat{f}(\pi'_i)) \mid f\in(\mathsf{T}(\Sigma),d)\to_1 (V,d_V)\right\}\\
&= L \sum_{i=1}^{n} \mathbf{K}_V(\pi_i,\pi'_i)
\end{aligned}
$$

with the first and last step follow by Definition of $\mathbf{K}_V$, second and second last step follow by Definition of $\hat{f}$, and steps 4 by linearity of d_V, step 5 by the fact that f is 1-Lipschitz and d_V is linear, step 6 by the fact that g is L-Lipschitz, and step 9 again by linearity of d_V. □

4.8.2 Lipschitz continuity and q-non-extensiveness

In this section we assume a PTSS $P=(\Sigma,A,R)$ and a Lipschitz factor assignments $L\colon \Sigma\to \mathbb{R}^{\infty}_{\geq 0}$, i.e. we assume Notation 4.17 with Σ_1 an empty signature and $\Sigma_2=\Sigma$. Moreover, we assume an arbitrary $\lambda\in(0,1]$.

Theorem 4.73. *Assume that (Σ,A,R) is a (L,λ)-Lipschitz PTSS, and let $\mathbf{D}\colon [0,1]^{\mathcal{T}(\Sigma)\times\mathcal{T}(\Sigma)}\to [0,1]^{\Delta(\mathcal{T}(\Sigma))\times\Delta(\mathcal{T}(\Sigma))}$ be any function that lifts 1-bounded pseudometrics on $\mathcal{T}(\Sigma)$ to 1-bounded pseudometrics on $\Delta(\mathcal{T}(\Sigma))$. If $\mathbf{D}$ satisfies Proposition 2.32 and Corollary 2.34 (where the functional $\mathbf{K}$ is replaced by $\mathbf{D}$), then any operator $f\in\Sigma$ with $L(f)<\infty$ is $L(f)$-Lipschitz continuous w.r.t. the $\mathbf{D}$-induced λ-bisimilarity metric $\mathbf{d}_{\mathbf{D},\lambda}$.*

Proof. First we note that Lemma 4.21 and Lemma 4.22 holds also with functional $\mathbf{K}$ replaced by $\mathbf{D}$. This can be viewed by observing that the only properties on $\mathbf{K}$ used in the proofs of these lemmas are that $\mathbf{K}$ satisfies Proposition 2.32 and Corollary 2.34.

The reasoning follows now the same line of argumentation as the proof of Theorem 4.5 and Theorem 4.19. Let d be the Lipschitz congruence closure (Definition 4.20) of the $\mathbf{D}$-induced λ-bisimilarity metric $\mathbf{d}_{\mathbf{D},\lambda}$ w.r.t. Σ_1,Σ_2,L. It is enough to show that d is a prefixed

point of $\mathbf{B}_{\mathbf{D},\lambda}$ on the lattice $([0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)}, \sqsubseteq)$, namely d satisfies the transfer condition of bisimulation metric

$$\forall (s,a,\pi) \in \longrightarrow . \exists (t,a,\pi') \in \longrightarrow . \ \lambda \cdot \mathbf{D}(d)(\pi,\pi') \leq d(s,t) \tag{4.28}$$

for all terms $s,t \in \mathsf{T}(\Sigma)$ with $d(s,t) < 1$.

If s and t have different outermost function symbols, then Equation 4.28 follows as in the case of Theorem 4.5 and Theorem 4.19.

Hence, it remains to show that for any given open term $t \in \mathbb{T}(\Sigma)$ and closed substitutions $\underline{\sigma}, \underline{\sigma}'$ with $\underline{\sigma}(x)$ and $\underline{\sigma}'(x)$ having different outermost function symbols for all $x \in \mathsf{Var}(t)$, the transfer condition of bisimulation metric (Equation 4.28) is satisfied for terms $\underline{\sigma}(t)$ and $\underline{\sigma}'(t)$. We will show this by structural induction over t.

The base case $t = x$ is trivial since it coincides with the case before where terms s and t, i.e. $\underline{\sigma}(x)$ and $\underline{\sigma}'(x)$, have different outermost function symbols.

The induction step $t = f(t_1,\ldots,t_n)$ requires that we distinguish two subcases. The first subcase $d(\underline{\sigma}(t),\underline{\sigma}'(t)) = \mathbf{d}(\underline{\sigma}(t),\underline{\sigma}'(t))$ (first argument of the min operator in Definition 4.6) is trivial and follows precisely from the earlier argumentation where s and t had different outermost function symbols.

The second remaining subcase is $d(\underline{\sigma}(t),\underline{\sigma}'(t)) = L(f)\sum_{i=1}^{n} d(\underline{\sigma}(t_i),\underline{\sigma}'(t_i))$ (second argument of the min operator in Definition 4.6 and, then, case $f \in \Sigma_2 \setminus \Sigma_1$ in Definition 4.20). Assume $d(\underline{\sigma}(t),\underline{\sigma}'(t)) < 1$. Let σ be any closed substitution with $\sigma(x_i) = \underline{\sigma}(t_i)$ and r be any (L,λ)-Lipschitz rule defining operator f with $\theta = \mathrm{trgt}(r)$ such that the transition $\sigma(f(x_1,\ldots,x_n)) \xrightarrow{a} \sigma(\theta)$ is derivable from r by σ. Like in the proof of Theorem 4.19 we construct an appropriate closed substitution σ' with $\sigma'(x_i) = \underline{\sigma}'(t_i)$ such that a transition $\sigma'(f(x_1,\ldots,x_n)) \xrightarrow{a} \sigma'(\theta)$ can be derived from r by σ' for which the metric transfer condition

$$\lambda \cdot \mathbf{D}(d)(\sigma(\theta),\sigma'(\theta)) \leq d(\sigma(f(x_1,\ldots,x_n)),\sigma'(f(x_1,\ldots,x_n))) \tag{4.29}$$

holds. In fact, we get by Lemma 4.22 (with $\mathbf{K}$ replaced by $\mathbf{D}$) the stricter statement

$$\mathbf{D}(d)(\sigma(\theta),\sigma'(\theta)) \leq \sum_{x\in\mathcal{V}_s} (d(\sigma(x),\sigma'(x)) \cdot \mathrm{Var}(x,\theta)) + \sum_{\mu\in\mathcal{V}_d} (\mathbf{D}(d)(\sigma(\mu),\sigma'(\mu)) \cdot \mathrm{Var}(\mu,\theta)$$

which implies Equation 4.29 since

$$\begin{aligned}
&\lambda \cdot \mathbf{D}(d)(\sigma(\theta),\sigma'(\theta)) \\
&\leq \sum_{x\in\mathcal{V}_s} \lambda \cdot d(\sigma(x),\sigma'(x)) \cdot \mathrm{Var}(x,\theta) + \sum_{\mu\in\mathcal{V}_d} \lambda \cdot \mathbf{D}(d)(\sigma(\mu),\sigma'(\mu)) \cdot \mathrm{Var}(\mu,\theta) \\
&= \sum_{i\in I} \left(\lambda \cdot d(\sigma(x_i),\sigma'(x_i)) \cdot \mathrm{Var}(x_i,\theta) + \sum_{\mu\in\mathrm{der}(r,x_i)} \lambda \cdot \mathbf{D}(d)(\sigma(\mu),\sigma'(\mu)) \cdot \mathrm{Var}(\mu,\theta) \right) \\
&\leq \sum_{i\in I} \left(\lambda \cdot d(\sigma(x_i),\sigma'(x_i)) \cdot \mathrm{Var}(x_i,\theta) + \sum_{\mu\in\mathrm{der}(r,x_i)} d(\sigma(x_i),\sigma'(x_i)) \cdot \mathrm{Var}(\mu,\theta) \right) \\
&= \sum_{i\in I} d(\sigma(x_i),\sigma'(x_i)) \left(\lambda \cdot \mathrm{Var}(x_i,\theta) + \sum_{\mu\in\mathrm{der}(r,x_i)} \mathrm{Var}(\mu,\theta) \right)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i \in I} d(\sigma(x_i), \sigma'(x_i)) \cdot \mathsf{copy}(r, \lambda, x_i) \\
&\leq \sum_{i \in I} d(\sigma(x_i), \sigma'(x_i)) \cdot L(f) \\
&= L(f) \sum_{i \in I} d(\sigma(x_i), \sigma'(x_i)) \\
&= d(\sigma(f(x_1, \ldots, x_n)), \sigma'(f(x_1, \ldots, x_n)))
\end{aligned}
$$

with step 2 by the fact that the only variables in the target θ of the PGSOS rule r are source variables and their derivatives, step 3 by the fact that $\sigma(x_i)$ and $\sigma'(x_i)$ satisfy the λ-bisimulation metric transfer condition, step 5 by definition of copy, step 6 by property $\mathsf{copy}(r, \lambda, x_i) \leq L(f)$ satisfied by the (L, λ)-Lipschitz rule r, and step 8 from the assumption $d(\underline{\sigma}(t), \underline{\sigma'}(t)) = L(f) \sum_{i=1}^{n} d(\underline{\sigma}(t_i), \underline{\sigma'}(t_i))$ and the equalities $\underline{\sigma}(t_i) = \sigma(x_i)$, $\underline{\sigma'}(t_i) = \sigma'(x_i)$, $\underline{\sigma}(t) = \sigma(f(x_1, \ldots, x_n))$ and $\underline{\sigma'}(t) = \sigma'(f(x_1, \ldots, x_n))$.

Hence, the transition $\sigma(f(x_1, \ldots, x_n)) \xrightarrow{a} \sigma(\mathrm{trgt}(r))$ derived from the f-defining rule r can be mimicked by $\sigma'(f(x_1, \ldots, x_n)) \xrightarrow{a} \sigma'(\mathrm{trgt}(r))$ (derived by the same r) such that the metric bisimulation transfer condition holds. Thus, operator f is $L(f)$-Lipschitz continuous w.r.t. d. □

Theorem 4.74. *Assume that (Σ, A, R) is a (L, λ)-Lipschitz PTSS and let $\mathbf{D} \colon [0,1]^{T(\Sigma) \times T(\Sigma)} \to [0,1]^{\Delta(T(\Sigma)) \times \Delta(T(\Sigma))}$ be any function that lifts 1-bounded pseudometrics on $T(\Sigma)$ to 1-bounded pseudometrics on $\Delta(T(\Sigma))$. If $\mathbf{D}$ satisfies Proposition 2.32 and Corollary 2.34 (where the functional $\mathbf{K}$ is replaced by $\mathbf{D}$), then each n-ary operator $f \in \Sigma$ with $L(f) \leq n^{(1/q)-1}$ for some $q \in [1, \infty]$ is q-non-extensive w.r.t. the $\mathbf{D}$-induced λ-bisimilarity metric $\mathbf{d}_{\mathbf{D},\lambda}$.*

Proof. Directly by Theorem 4.73 and $n^{(1/q)-1} \sum_{i=1}^{n} d(t_i, t_i') \leq \left(\sum_{i=1}^{n} d(t_i, t_i')^q \right)^{1/q}$. □

4.8.3 Uniform continuity

The following results are the anologous to Proposition 4.32 and Theorem 4.33.

Proposition 4.75. *Let $\mathbf{D} \colon [0,1]^{T(\Sigma) \times T(\Sigma)} \to [0,1]^{\Delta(T(\Sigma)) \times \Delta(T(\Sigma))}$ be monotone and $s, t \in T(\Sigma)$ be arbitrary closed terms. Then*

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

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

Proof. The same arguments used in the proof of Proposition 4.32 with $\mathbf{K}$ replaced by $\mathbf{D}$ apply. This can be viewed by observing that the only property of functional $\mathbf{K}$ used in the proof of Proposition 4.32 is monotonicity. □

Theorem 4.76. *Assume $\lambda < 1$. Let $\mathbf{D} \colon [0,1]^{T(\Sigma) \times T(\Sigma)} \to [0,1]^{\Delta(T(\Sigma)) \times \Delta(T(\Sigma))}$ be any function that lifts 1-bounded pseudometrics on $T(\Sigma)$ to 1-bounded pseudometrics on $\Delta(T(\Sigma))$. If $\mathbf{D}$ is monotone, then if an operator $f \in \Sigma$ is Lipschitz continuous w.r.t. $\mathbf{d}_{\mathbf{D},\lambda,k}$ for each $k \in N$, then f is uniformly continuous w.r.t. $\mathbf{d}_{\mathbf{D},\lambda}$.*

Proof. The same arguments used in the proof of Theorem 4.33 apply. This can be viewed by observing that the only property of $\mathbf{d}_k$ and $\mathbf{d}$ used in that proof are Proposition 4.32 and $\mathbf{d}_k \sqsubseteq \mathbf{d}$, which are now given by Proposition 4.75 and $\mathbf{d}_{\mathbf{D},\lambda,k} \sqsubseteq \mathbf{d}_{\mathbf{D},\lambda}$. □

For the remainder of this section we assume a PTSS $P = (\Sigma, A, R)$ and a Lipschitz factor assignments $L : (\mathbb{N} \times \Sigma) \to \mathbb{R}^{\infty}_{\geq 0}$, i.e. we assume Notation 4.36 with Σ_1 an empty signature and $\Sigma_2 = \Sigma$. Moreover, we assume an arbitrary $\lambda \in (0, 1]$.

Theorem 4.77. *Let (Σ, A, R) be a (L, λ)-uniformly continuous PTSS and $\mathbf{D} : [0,1]^{T(\Sigma)\times T(\Sigma)} \to [0,1]^{\Delta(T(\Sigma))\times\Delta(T(\Sigma))}$ be any function that lifts 1-bounded pseudometrics on $T(\Sigma)$ to 1-bounded pseudometrics on $\Delta(T(\Sigma))$. If $\mathbf{D}$ satisfies Proposition 2.32 and Corollary 2.34 (where the functional $\mathbf{K}$ is replaced by $\mathbf{D}$), then, given any operator $f \in \Sigma$, if $L_k(f) < \infty$ for all $k \in \mathbb{N}$, then*

1. *f is $L_k(f)$-Lipschitz continuous w.r.t. $\mathbf{d}_{\mathbf{D},\lambda,k}$ for any $k \in \mathbb{N}$, and*
2. *f is uniformly continuous w.r.t. $\mathbf{d}_{\mathbf{D},\lambda}$ if $\lambda < 1$.*

Proof. The same argument of the proof of Theorem 4.77 apply. In that proof we constructed from the PTSS $P_2 = (\Sigma_2, A, R_2)$ a PTSS P_2' such each term $t \in \mathsf{T}(\Sigma_2)$ has for any $k \in \mathbb{N}$ a corresponding term $t_k \in \mathsf{T}(\Sigma_2')$ that behaves for the first k steps as t and stops afterwards. Hence, if f is $L_k(f)$-Lipschitz continuous w.r.t. $\mathbf{d}_k$ in the induced model of P_2, then f_k is $L_k(f)$-Lipschitz continuous w.r.t. $\mathbf{d}$ in the induced model of P_2'. This reduces the proof of uniform continuity to the case of Lipschitz continuity (Theorem 4.33 and Theorem 4.19). Analogously, if f is $L_k(f)$-Lipschitz continuous w.r.t. $\mathbf{d}_{\mathbf{D},\lambda,k}$ in the induced model of P_2, then f_k is $L_k(f)$-Lipschitz continuous w.r.t. $\mathbf{d}_{\mathbf{D},\lambda}$ in the induced model of P_2'. This reduces the proof of uniform continuity to the case of Lipschitz continuity (Theorem 4.76 and Theorem 4.73). □

4.9 Closing remarks

We developed SOS specification formats for an expressive spectrum of compositionality properties. The explored compositionality properties capture all important process algebra operators and many operators of programming languages. The formats allow us to simultaneously specify operators with different compositionality properties. Our fundamental insight is that the modulus of continuity of an operator is closely related to how many times this operator replicates its arguments. Hence, the formats restrict the replication of processes in the specification rules according to the modulus of continuity of the compositionality property.

The rule and specification formats become less restrictive (i.e. allow a larger class of specifications) when a less demanding (i.e. weaker) compositionality property is considered. The formats for Lipschitz continuous and uniformly continuous operators exploit the compositionality guarantees provided by non-extensive operators in order to admit a wide class of specifications. The rule and specification formats also give insight into the interplay between the replication of processes, probabilistic choices between processes, and the (step) discount of the bisimulation metric. Our format and results pave the way for a robust and modular approach to specify and verify probabilistic systems using probabilistic process algebras and probabilistic programming languages.

We remark that our rule formats do not satisfy a completeness property, i.e. one may define non-extensive, Lipschitz continuous or uniformly continuous operators specified by rules that do not match our format. This does not come as a surprise since none of the SOS rule formats developed so far (since the late 1980s) is complete. However, we want to stress that for all process algebra operators discussed in this chapter (and many other specifications of standard programming language operators) it holds that if the specification does not satisfy a rule format w.r.t. some compositionality properties, then the specified operator will actually also not satisfy that compositionality property.[5]

Following the development and analysis of weak behavioral metrics as proposed in Section 3.5 the logical next step would be to develop SOS specification formats for the spectrum of compositionality properties (Figure 4.1) w.r.t. those weak behavioral metrics (and similarly also for behavioral metrics w.r.t. trace [AFS04; FL14] and test based semantics). As a first step we need to develop SOS rule formats for the respective notion of probabilistic behavioral equivalence. Preliminary results suggest that (at least for PGSOS rules) the format restrictions on GSOS rules (that describe nondeterministic LTSs as operational models), e.g. formats of [BFG04; MRG07], can be lifted directly to the probabilistic setting. As second step we need to develop SOS rule and specification formats for the respective notion of probabilistic behavioral metric. We expect that the rule restrictions on the process replication behavior induced by the various compositionality properties apply orthogonally to the rule restrictions that are induced by the respective behavioral equivalence (kernel of the behavioral metric).

Languages for probabilistic programming (cf. [Gor+14]) provide, additionally to the usual functional or imperative programming constructs, constructs (a) to draw values at random from distributions, and (b) to condition values of variables via observations. In order to model those languages in the SOS framework we need to extend the language of distribution terms. The language introduced in Definition 2.5 allows us to express (unconditional) probabilistic choice between distributions and (unconditional independent) product of distributions. The first construct (a) requires to separate the draw of values from a distribution and the construction of terms. First we provide an example showing that the distribution terms of Definition 2.5 cannot express a single draw from a distribution and multiple applications of that draw to different positions in a term. Consider the rule

$$\frac{x \xrightarrow{a} \mu}{f(x) \xrightarrow{a} g(\mu,\mu)}$$

If $s \xrightarrow{a} 0.5\delta(t_1) + 0.5\delta(t_2)$, then the rule allows to derive $f(s) \xrightarrow{a} 0.25\delta(g(t_1,t_1)) + 0.25\delta(g(t_1,t_2)) + 0.25\delta(g(t_2,t_1)) + 0.25\delta(g(t_2,t_2))$. In other words, the semantics of the distribution term $g(\mu,\mu)$ is to draw twice independently values $y_1 \in \{t_1,t_2\}$ and $y_2 \in \{t_1,t_2\}$ from μ and then compose the respective term $g(y_1,y_2)$. There is no rule that allows us to derive from the transition $s \xrightarrow{a} 0.5\delta(t_1)+0.5\delta(t_2)$ the transition $f(s) \xrightarrow{a} 0.5\delta(g(t_1,t_1)) + 0.5\delta(g(t_2,t_2))$, i.e. only one draw y from μ and then compose $f(y,y)$.

Motivated by [MS13, Definition 2] we propose to extend the language of distribution terms by the additional case $\{\mu \rightsquigarrow x\}\theta \in \mathbb{DT}(\Sigma)$ if $\theta \in \mathbb{DT}(\Sigma)$ with semantics $\sigma(\{\mu \rightsquigarrow x\}\theta)(t) = \sum_{s\in\mathsf{T}(\Sigma)} \sigma(\mu)(s) \cdot (\sigma \circ [s/x])(\theta)(t)$ with $[s/x]$ denoting the substitution $[s/x](x) = s$ and $[s/x](y) = y$ if $x \neq y$. In fact, $\{\mu \rightsquigarrow x\}\theta$ allows us to express

[5] This property supports our claim that the rule formats are expressive.

any $f(\theta_1, \ldots, \theta_n)$ by defining a new term θ' where each occurrence of any distribution variable μ_j in any of the θ_i is replaced by some fresh $\delta(x_j)$ (i.e. θ' is a convex combination of distribution variables and Dirac embeddings of state terms). Then the semantics of $\{\mu_1 \rightsquigarrow x_1\} \ldots \{\mu_m \rightsquigarrow x_m\}\theta'$ (with m the number of occurrences of distribution variables in θ) coincides with $f(\theta_1, \ldots, \theta_n)$. The second construct to express conditional probabilities would require an expression like $\theta|C$ with C a boolean formula built from atomic propositions $x \xrightarrow{a}$ and $x \not\xrightarrow{a}$, with $\sigma(\theta|C)(t) = \sigma(\theta)(t)/\sigma(\theta)(\{s \in \mathsf{T}(\Sigma) \mid s \models C)$ if $\sigma(\theta)(\{s \in \mathsf{T}(\Sigma) \mid s \models C) > 0$ and $\sigma(\theta|C)(t)$ undefined otherwise (i.e. the transition with $\theta|C$ in the rule target cannot be derived), whereby $s \models x \xrightarrow{a}$ iff $s \xrightarrow{a}$, $s \models x \not\xrightarrow{a}$ iff $s \not\xrightarrow{a}$, and boolean connectives as usual. We leave the in-depth exploration of this extension and their properties as future work.

As another research direction we propose to investigate the distance between operators (instead of closed terms) to describe the behavioral distance whenever one operator needs to be replaced or approximated by another. Intuitively, if an operator becomes unavailable the distance between operators will suggest an optimal replacement operator to build an alternative system which is closest to the original system. The distance between operators $\mathbf{d}_\Sigma \colon \Sigma \times \Sigma \to [0,1]$ could be defined as

$$\mathbf{d}_\Sigma(f,g) = \begin{cases} \sup\limits_{t_1,\ldots,t_n \in \mathsf{T}(\Sigma)} \mathbf{d}(f(t_1,\ldots,t_n), g(t_1,\ldots,t_n)) & \text{if } \mathsf{r}(f) = \mathsf{r}(g) \\ 1 & \text{otherwise} \end{cases}$$

The distance $\mathbf{d}_\Sigma$ lifts to open terms in the obvious way. An interesting question is now to determine compositionality properties of this pseudometric and to decide what is the optimal replacement context t for a given operator f.

Chapter 5

A denotational model of metric compositionality

5.1 Introduction

In the former chapters we explored the behavioral semantics of compositional operators (Chapter 3) and developed a specification theory for compositional operators (Chapter 4). Now we will develop a denotational model for the compositionality properties of operators by formalizing the essential structure that determines the modulus of continuity of an operator. The denotation of an operator arises by composing the denotations of the primitive operators such as action prefix, sequencing, probabilistic and nondeterministic choice (similar to the construction of basic process algebra expressions that mimic the operational semantics of an operator [ABV94]). The domain model will not only provide a deeper understanding of the essential nature of metric compositionality but allows also to analyze for each operator which aspects of its behavior (e.g. probabilistic choice vs. process replication) contribute how much to its overall compositionality property.

We start by developing in Section 5.2 for a concrete process algebra an appropriate denotational model. The denotation of an open process term describes for each resolution of the nondeterministic choices how many instances of each process variable are spawned while the process evolves. The number of spawned process replicas is weighted by the likelihood of its realization just like the bisimulation metric weights the distance between target states by their reachability. We derive from the denotation of an open process term an upper bound on the bisimulation distance between the closed instances of the denoted process. In Section 5.3 we generalize this method to arbitrary processes whose operational semantics is specified by probabilistic GSOS rules. The denotation of an open process term is derived from the rules that specify the operational semantics of the operators used in the process term. In detail, we define a functional that computes from a denotation of an open process term w.r.t. up-to-n bisimulation metric the denotation of this process term w.r.t. up-to-$(n+1)$ bisimulation metric. The least fixed point of this functional defines then the denotation of an open process term w.r.t. bisimilarity metric. The denotation of an open process terms allows to derive an upper bound on the bisim-

ulation distance between closed instances of that term. Section 5.4 applies those results and derives the modulus of continuity of operators and open terms from their respsective denotation. In fact, the upper bound on the bisimulation distance between closed instances of $f(x_1, \ldots, x_{r(f)})$ is a modulus of continuity of operator f if the denotation of $f(x_1, \ldots, x_{r(f)})$ is finitely bounded. In this case the operator f is uniformly continuous and allows for compositional reasoning.

On the one hand, the denotation allows us to study the composition of operators, temporal invariance of compositionality properties, and to define a set of basic operators that allow to describe operators of any compositionality property. On the other hand, the denotational model opens also the door for a new approach to derive SOS rule and specification formats for any given compositionality property. While the specification formats in Chapter 4 were derived from an extensive analysis of various rule patterns (similar to the development of the GSOS format [BIM95] and the *ntyft*/*ntyxt* [Gro93] format) we derive now specification formats from the denotation that describe the essential nature of the compositionality of the denoted operators.

The main contributions of this chapter are:

1. We develop a denotational model for a concrete process algebra (Section 5.2) and generalize this model and the computation method to arbitrary languages (Section 5.3)

2. We derive from the denotation of operators and terms an upper bound on the bisimulation distance between the closed instances of the denoted process (Definitions 5.3, 5.13, 5.21, and Theorem 5.45).

3. We show that the denotation of basic process algebra operators form a basis for the finite denotational model (similar to the fact the operational semantics of basic process algebra operators forms a basis for finite PTS) (Table 5.1, Theorem 5.46).

4. We derive from the denotation of operators (resp. terms) a specification format of uniformly continuous operators (Theorem 5.55, resp. Theorem 5.61)

5. We derive from any given modulus of continuity an appropriate denotation s.t. operators (resp. terms) satisfying the denotation admit that modulus of continuity (Theorem 5.57, resp. Theorem 5.64).

This chapter has been partially published as [GT13; GT14]. We developed in [GT13] the fixed point approach to compute the upper bound on the distance between processes and in [GT14] formalized the underlying denotational model. The denotational model separates clearly between nondeterministic choice, probabilistic choice, and process replication. This chapter extends significantly the earlier published material as follows:

- We consider now discounted and nondiscounted bisimulation metric with a step discount $\lambda \in (0, 1]$ whereby in [GT14] we considered only nondiscounted bisimulation metric. In essence, the discount factor λ weights the multiplicities by λ^n if the subprocesses are replicated just after performing n steps.

- We allow now that the variables in open processes are substituted by processes that may disagree completely, i.e. that have a bisimulation distance of 1.

- We provide now a detailed discussion about the connection between operational behavior, process replication and bisimulation distance (section 5.3.4).

- We extend the computation of the modulus of continuity of operators [GT13; GT14] to open terms and derive from the metric bisimulation invariance condition a corresponding invariance condition of the modulus of continuity.

5.2 Denotational model

We will develop in this section a denotational model for open terms. Essentially, the denotation of an open term t describes for each variable x occurring in t how many copies of x are spawned while t evolves. The denotation of t will allows us to formulate an upper bound on the bisimulation distance between the closed instances of t, given the bisimulation distance between the closed instances of the variables occurring in t. In this section we consider a concrete process algebra. In the next sections we generalize our method to arbitrary PGSOS specifications.

Let Σ_{PA} be the signature of the core operators of the probabilistic process algebra in [DL12] defined by the stop process 0, a family of n-ary prefix operators $a.([q_1]\_ \oplus \cdots \oplus [q_n]\_)$ with $a \in A$, $n \geq 1$, $q_1, \ldots, q_n \in (0,1]$ and $\sum_{i=1}^{n} q_i = 1$, alternative composition $\_+\_$, and synchronous parallel composition $\_ \parallel \_$. We write $a.\bigoplus_{i=1}^{n}[q_i]\_$ for $a.([q_1]\_ \oplus \cdots \oplus [q_n]\_)$, and $a.\_$ for $a.[1]\_$ (deterministic prefix operator). The PTSS $P_{\mathrm{PA}} = (\Sigma_{\mathrm{PA}}, A, R_{\mathrm{PA}})$ is given by the following PGSOS rules in R_{PA} (same as in Table 3.1 and repeated here for convenience):

$$\frac{}{a.\bigoplus_{i=1}^{n}[q_i]x_i \xrightarrow{a} \sum_{i=1}^{n} q_i \delta(x_i)} \qquad \frac{x_1 \xrightarrow{a} \mu_1 \quad x_2 \xrightarrow{a} \mu_2}{x_1 \parallel x_2 \xrightarrow{a} \mu_1 \parallel \mu_2}$$

$$\frac{x_1 \xrightarrow{a} \mu_1}{x_1 + x_2 \xrightarrow{a} \mu_1} \qquad \frac{x_2 \xrightarrow{a} \mu_2}{x_1 + x_2 \xrightarrow{a} \mu_2}$$

We call the open terms in $\mathbb{T}(\Sigma_{\mathrm{PA}})$ *nondeterministic probabilistic process terms*. We define two important subclasses of $\mathbb{T}(\Sigma_{\mathrm{PA}})$ that allow for a simpler approximation of the distance between the closed instances of open terms. Let $\mathbb{T}_{\mathrm{det}}(\Sigma_{\mathrm{PA}})$ be the set of *deterministic process terms*, which are those terms of $\mathbb{T}(\Sigma_{\mathrm{PA}})$ that are built exclusively from the stop process 0, deterministic prefix $a.\_$, and synchronous parallel composition $\_ \parallel \_$ (no nondeterministic and no probabilistic choices). We call the open terms in $\mathbb{T}_{\mathrm{det}}(\Sigma_{\mathrm{PA}})$ deterministic because all probabilistic or nondeterministic choices in the operational semantics of the closed instances $\sigma(t)$, with $\sigma \colon \mathcal{V}_s \to \mathsf{T}(\Sigma_{\mathrm{PA}})$ any closed substitution, arise exclusively from the processes in σ. Let $\mathbb{T}_{\mathrm{prob}}(\Sigma_{\mathrm{PA}})$ be the set of *probabilistic process terms*, which are those terms in $\mathbb{T}(\Sigma_{\mathrm{PA}})$ that are built exclusively from the stop process 0, probabilistic prefix $a.\bigoplus_{i=1}^{n}[q_i]\_$, and synchronous parallel composition $\_ \parallel \_$ (no nondeterministic choices). Again, all nondeterministic choices in $\sigma(t)$ arise exclusively from the processes in σ.

The denotational model for $\mathbb{T}_{\mathrm{det}}(\Sigma_{\mathrm{PA}})$ is developed in Section 5.2.1, the denotational model for $\mathbb{T}_{\mathrm{prob}}(\Sigma_{\mathrm{PA}})$ is developed in Section 5.2.2, and the denotational model for $\mathbb{T}(\Sigma_{\mathrm{PA}})$ is developed in Section 5.2.3.

5.2.1 Denotation of deterministic process terms

We start with introducing the notion of multiplicity. Multiplicities will be used as denotations for deterministic processes terms in $\mathbb{T}_{\text{det}}(\Sigma_{\text{PA}})$.

Definition 5.1. A *multiplicity* is a mapping $m\colon \mathcal{V} \to \mathbb{R}^{\infty}_{\geq 0}$. The set of all multiplicities is denoted by $\mathcal{M}$.

The denotation of a deterministic process term $t \in \mathbb{T}_{\text{det}}(\Sigma_{\text{PA}})$ is a multiplicity in $\mathcal{M}$, denoted $[\![t]\!]_{\mathcal{M}}$, that describes for each process variable $x \in \mathcal{V}$ how many copies of x or some derivative of x are spawned while t evolves. The number of copies that are delayed or spawned after the process evolved are discounted. Therefore, the multiplicity $[\![t]\!]_{\mathcal{M}}(x)$ of variable x is a non-negative extended real $[\![t]\!]_{\mathcal{M}}(x) \in \mathbb{R}^{\infty}_{\geq 0}$ and not only a natural number. The multiplicities $[\![t]\!]_{\mathcal{M}}$ are defined inductively over the structure of t by

$$[\![t]\!]_{\mathcal{M}}(x) = \begin{cases} 0 & \text{if } t = 0 \\ 1 & \text{if } t = x \\ 0 & \text{if } t = y \in \mathcal{V} \text{ and } y \neq x \\ [\![t_1]\!]_{\mathcal{M}}(x) + [\![t_2]\!]_{\mathcal{M}}(x) & \text{if } t = t_1 \parallel t_2 \\ \lambda \cdot [\![t']\!]_{\mathcal{M}}(x) & \text{if } t = a.t' \end{cases}$$

for all $x \in \mathcal{V}$.

We use notation $0 \in \mathcal{M}$ for the multiplicity that assigns 0 to each $x \in \mathcal{V}$, and $n_V \in \mathcal{M}$ with $V \subseteq \mathcal{V}$ for the multiplicity such that $n_V(x) = n$ if $x \in V$ and $n_V(x) = 0$ if $x \notin V$. We write n_x for $n_{\{x\}}$. As it will become clear in the next sections, we need the denotation $m(x) = \infty$ for (unbounded) recursion and replication.

We will approximate the bisimulation distance between $\sigma_1(t)$ and $\sigma_2(t)$ for closed substitutions σ_1, σ_2 using the denotation of t and the bisimulation distance between processes $\sigma_1(x)$ and $\sigma_2(x)$ of variables $x \in \mathsf{Var}(t)$.

Definition 5.2. A *process distance* is a mapping $e\colon \mathcal{V} \to [0, \lambda] \cup \{1\}$. The set of all process distances is denoted by $\mathcal{E}$.

Each pair of substitutions σ_1, σ_2 induces a process distance. We define the *process distance induced by closed substitutions* σ_1, σ_2 as $\mathbf{d}(\sigma_1, \sigma_2) \in \mathcal{E}$ defined by

$$\mathbf{d}(\sigma_1, \sigma_2)(x) = \mathbf{d}(\sigma_1(x), \sigma_2(x))$$

for all $x \in \mathcal{V}$.

Definition 5.3. The *deterministic distance approximation from above* $\mathbf{D}\colon (\mathcal{M} \times \mathcal{E}) \to [0, \lambda] \cup \{1\}$ is defined by

$$\mathbf{D}(m, e) = \begin{cases} \lambda\left(1 - \prod_{x \in \mathcal{V}} \left(1 - \frac{e(x)}{\lambda}\right)^{m(x)}\right) & \text{if } \forall x \in \mathcal{V}.e(x) < 1 \\ 1 & \text{if } \exists x \in \mathcal{V}.e(x) = 1 \end{cases}$$

for all $m \in \mathcal{M}$ and $e \in \mathcal{E}$, with definition[1] $1^{\infty} = 1$ and $0^0 = 1$.

[1]The expressions 1^{∞} and 0^0 are in general indeterminate. However, in our application context we assign a value that reflects the meaning of the respective approximation functionals.

To understand the functional $\mathbf{D}$ in case $e(x) < 1$ for all $x \in \mathcal{V}$, assume $e = \mathbf{d}(\sigma_1, \sigma_2)$. Recall that $e(x)$ is the distance between processes $\sigma_1(x)$ and $\sigma_2(x)$, and $e(x) \leq \lambda$. In other words, the processes $\sigma_1(x)$ and $\sigma_2(x)$ disagree by at most $e(x)/\lambda$ on their behavior after performing their initial actions. Hence, $\sigma_1(x)$ and $\sigma_2(x)$ agree by at least $1-e(x)/\lambda$. Thus, $m(x)$ copies of $\sigma_1(x)$ and $m(x)$ copies of $\sigma_2(x)$ agree after performing their initial actions by at least $(1-e(x)/\lambda)^{m(x)}$. Then the instances of all processes in $\mathcal{V}$ agree by at least $\prod_{x\in\mathcal{V}}(1-e(x)/\lambda)^{m(x)}$, and disagree by at most $1-\prod_{x\in\mathcal{V}}(1-e(x)/\lambda)^{m(x)}$. Finally, $m(x)$ copies of $\sigma_1(x)$ and of $\sigma_2(x)$ disagree initially by at most $\lambda(1-\prod_{x\in\mathcal{V}}(1-e(x)/\lambda)^{m(x)}) = \mathbf{D}(m,e)$.

Example 5.4. Consider the deterministic process term $t = x \parallel x$ and the substitutions $\sigma_1(x) = a.a.0$ and $\sigma_2(x) = a.([0.9]a.0 \oplus [0.1]0)$. In this and all following examples we assume that σ_1 and σ_2 coincide on all other variables for which the substitution is not explicitly defined, i.e. $\sigma_1(y) = \sigma_2(y)$ if $y \neq x$ in this example. It is clear that $\mathbf{d}(\sigma_1(x), \sigma_2(x)) = 0.1\lambda$, which is the likelihood that $\sigma_2(x)$ can perform the action a only once, weighted by the discount λ. Then, $\mathbf{d}(\sigma_1(t), \sigma_2(t)) = 0.19\lambda$, which is the likelihood that either the first argument of the parallel composition $\sigma_2(x \parallel x)$, or the second argument of the parallel composition $\sigma_2(x \parallel x)$, or both arguments of the parallel composition $\sigma_2(x \parallel x)$ can perform the action a only once, weighted by the discount λ. The denotation of t is $[\![t]\!]_{\mathcal{M}}(x) = 2$, namely $[\![t]\!]_{\mathcal{M}} = 2_x$. Then, $\mathbf{D}([\![t]\!]_{\mathcal{M}}, \mathbf{d}(\sigma_1, \sigma_2)) = \lambda(1-(1-0.1\lambda/\lambda)^2) = 0.19\lambda = \mathbf{d}(\sigma_1(t), \sigma_2(t))$.

Example 5.5. Consider the deterministic process term $t = a.x$ and the substitutions $\sigma_1(x) = a.a.0$ and $\sigma_2(x) = a.([0.9]a.0 \oplus [0.1]0)$ already considered in Example 5.4 with $\mathbf{d}(\sigma_1(x), \sigma_2(x)) = 0.1\lambda$. Then $\mathbf{d}(\sigma_1(t), \sigma_2(t)) = 0.1\lambda^2$. The multiplicity of t is $[\![t]\!]_{\mathcal{M}}(x) = \lambda$, namely $[\![t]\!]_{\mathcal{M}} = (\lambda)_x$. Then, $\mathbf{D}((\lambda)_x, \mathbf{d}(\sigma_1, \sigma_2)) = \lambda(1-(1-0.1)^{\lambda}) \geq \lambda(1-(1-0.1\lambda)) = 0.1\lambda^2 = \mathbf{d}(\sigma_1(t), \sigma_2(t))$ (using Bernoulli's inequality $(1-0.1)^{\lambda} \leq 1-0.1\lambda$).

Consider the deterministic process term $t = a.(x \parallel x)$ and the same substitutions σ_1, σ_2. Then $\mathbf{d}(\sigma_1(t), \sigma_2(t)) = \lambda^2(1-(1-0.1)^2) = 0.19\lambda^2$. The multiplicity of t is $[\![t]\!]_{\mathcal{M}}(x) = 2\lambda$, i.e. $[\![t]\!]_{\mathcal{M}} = (2\lambda)_x$. The deterministic approximation functional gives $\mathbf{D}((2\lambda)_x, \mathbf{d}(\sigma_1, \sigma_2)) = \lambda(1-(1-0.1)^{2\lambda}) = \lambda(1-(1-0.1)^{\min(2\lambda,1)}(1-0.1)^{\max(2\lambda-1,0)}) \geq \lambda(1-(1-0.1\min(2\lambda,1))(1-0.1\max(2\lambda-1,0))) \geq 0.2\lambda^2 - 0.01\lambda\min(2\lambda,1)\max(2\lambda-1,0)$. Since $0.01\lambda\min(2\lambda,1)\cdot\max(2\lambda-1,0) \leq 0.01\lambda\cdot 1\cdot\lambda = 0.01\lambda^2$, we conclude $\mathbf{d}(\sigma_1(t), \sigma_2(t)) \leq \mathbf{D}((2\lambda)_x, \mathbf{d}(\sigma_1, \sigma_2))$.

The functional $\mathbf{D}$ defines an upper bound on the bisimulation distance between deterministic processes.

Proposition 5.6. *Let $t \in \mathbb{T}_{det}(\Sigma_{PA})$ be a deterministic process term and σ_1, σ_2 be closed substitutions. Then $\mathbf{d}(\sigma_1(t), \sigma_2(t)) \leq \mathbf{D}([\![t]\!]_{\mathcal{M}}, \mathbf{d}(\sigma_1, \sigma_2))$.*

Proof. It is easy to verify that the denotations defined for $t \in \mathbb{T}_{\mathrm{det}}(\Sigma_{\mathrm{PA}})$ are exactly those that are computed by the least fixed point of $\mathbf{F}$ in Section 5.3. The Proposition follows then from Theorem 5.45 and the observation that $\mathbf{D}([\![t]\!]_{\mathcal{M}}, \mathbf{d}(\sigma_1, \sigma_2)) = \mathbf{A}([\![t]\!], \mathbf{d}(\sigma_1, \sigma_2))$. □

The distance $\mathbf{d}(\sigma_1, \sigma_2)$ abstracts from the concrete reactive behavior of terms $\sigma_1(x)$ and $\sigma_2(x)$. It is not hard to see that for deterministic process terms without parallel composition and non-discounted bisimulation metric the approximation functional $\mathbf{D}$ gives

the exact bisimulation distance. However, the parallel composition of processes may lead to an overapproximation if the bisimulation distance of process instances arises (at least partially) from reactive behavior on which the processes cannot synchronize.

Example 5.7. Consider the deterministic process term $t = x \parallel a.a.0$ and the substitutions $\sigma_1(x) = b.b.0$ and $\sigma_2(x) = b.([0.9]b.0 \oplus [0.1]0)$ with $\mathbf{d}(\sigma_1(x), \sigma_2(x)) = 0.1\lambda$. We have $\mathbf{d}(\sigma_1(t), \sigma_2(t)) = 0$ since both $\sigma_1(t)$ and $\sigma_2(t)$ cannot proceed. Note that the bisimulation distance between $\sigma_1(x)$ and $\sigma_2(x)$ arises from the difference on performing action b which cannot synchronize with a. The denotation of t is $[\![t]\!]_{\mathscr{M}}(x) = 1$ which gives in this case an overapproximation of the distance $\mathbf{d}(\sigma_1(t), \sigma_2(t)) = 0 < \mathbf{D}([\![t]\!]_{\mathscr{M}}, \mathbf{d}(\sigma_1, \sigma_2)) = \lambda(1-(1-0.1)) = 0.1\lambda$. However, for $\sigma_1'(x) = a.a.0$ and $\sigma_2'(x) = a.([0.9]a.0 \oplus [0.1]0)$ with $\mathbf{d}(\sigma_1'(x), \sigma_2'(x)) = 0.1\lambda$ we get $\mathbf{d}(\sigma_1'(t), \sigma_2'(t)) = 0.1\lambda = \mathbf{D}([\![t]\!]_{\mathscr{M}}, \mathbf{d}(\sigma_1', \sigma_2'))$.

We remark that the abstraction of the closed substitutions to process distances is intentional and very much in line with common compositionality criteria that relate the distance of composed processes with the distance of the process components.

We conclude this section by introducing an order over $\mathscr{M}$ and $\mathscr{E}$ such that the deterministic distance approximation functional $\mathbf{D}$ is monotone in both arguments.

Let $0 \in \mathscr{E}$ be the process distance with $0(x) = 0$ for all $x \in \mathscr{V}$. It is clear that $\mathbf{D}(m, e) = 0$ if $m = 0 \in \mathscr{M}$ or $e = 0 \in \mathscr{E}$.

Definition 5.8. We order the elements of $\mathscr{M}$ by $\sqsubseteq \subseteq \mathscr{M} \times \mathscr{M}$ defined as

$$m_1 \sqsubseteq m_2 \text{ iff } m_1(x) \leq m_2(x) \text{ for all } x \in \mathscr{V}$$

for all $m_1, m_2 \in \mathscr{M}$.

Proposition 5.9. *$(\mathscr{M}, \sqsubseteq)$ is a complete lattice with least element 0.*

Proof. It is not hard to see that $(\mathscr{M}, \sqsubseteq)$ is a partially ordered set with $(\inf M)(x) = \inf_{m \in M} m(x)$ and $(\sup M)(x) = \sup_{m \in M} m(x)$ for all $M \subseteq \mathscr{M}$ (by the fact that $(\mathbb{R}^{\infty}_{\geq 0}, \leq)$ is a complete lattice). Moreover, $\inf_{m \in \mathscr{M}} m(x) = 0 = 0(x)$. □

Definition 5.10. We order the elements of $\mathscr{E}$ by $\sqsubseteq \subseteq \mathscr{E} \times \mathscr{E}$ defined as

$$e_1 \sqsubseteq e_2 \text{ iff } e_1(x) \leq e_2(x) \text{ for all } x \in \mathscr{V}$$

for all $e_1, e_2 \in \mathscr{E}$.

Proposition 5.11. *Let $m, m' \in \mathscr{M}$ and $e, e' \in \mathscr{E}$. Then*

1. *$\mathbf{D}(m, e) \leq \mathbf{D}(m', e)$ if $m \sqsubseteq m'$;*
2. *$\mathbf{D}(m, e) \leq \mathbf{D}(m, e')$ if $e \sqsubseteq e'$.*

Proof. Consider first case 1. If $e(x) = 1$ for some $x \in \mathscr{V}$, then we have $\mathbf{D}(m, e) = 1 = \mathbf{D}(m', e)$. If $e(x) < 1$ for all $x \in \mathscr{V}$, then we have $e(x) \leq \lambda$ for all $x \in \mathscr{V}$ and, by exploiting $0 \leq 1 - e(x)/\lambda \leq 1$, we get

$$\mathbf{D}(m, e) = \lambda\left(1 - \prod_{x \in \mathscr{V}} (1 - e(x)/\lambda)^{m(x)}\right) \leq \lambda\left(1 - \prod_{x \in \mathscr{V}} (1 - e(x)/\lambda)^{m'(x)}\right) = \mathbf{D}(m', e).$$

Consider now case 2. Observe first that $\mathbf{D}(m,e) \leq 1$. Hence, if $e'(x) = 1$ for some $x \in \mathcal{V}$, then we have $\mathbf{D}(m,e) \leq 1 = \mathbf{D}(m',e)$. If $e'(x) < 1$ for all $x \in \mathcal{V}$, and therefore $e(x) < 1$ for all $x \in \mathcal{V}$, then we have

$$\mathbf{D}(m,e) = \lambda\left(1 - \prod_{x\in\mathcal{V}}(1-e(x)/\lambda)^{m(x)}\right) \leq \lambda\left(1 - \prod_{x\in\mathcal{V}}(1-e'(x)/\lambda)^{m(x)}\right) = \mathbf{D}(m,e').$$

□

5.2.2 Denotation of probabilistic process terms

Probabilistic multiplicities are distributions on the set of the multiplicities $\mathcal{M}$.

Definition 5.12. A *probabilistic multiplicity* is a distribution on $\mathcal{M}$. The set of all distributions on $\mathcal{M}$ is denoted by $\mathcal{P}$.

The denotation of a probabilistic process term $t \in \mathbb{T}_{\text{prob}}(\Sigma_{\text{PA}})$ is a distribution in $\mathcal{P}$, denoted $[\![t]\!]_{\mathcal{P}}$, that describes for each multiplicity $m \in \mathcal{M}$ the likelihood $[\![t]\!]_{\mathcal{P}}(m)$ that for each process variable $x \in \mathsf{Var}(t)$ exactly $m(x)$ copies of x or some derivative of x are spawned while t evolves. The probabilistic multiplicity $[\![t]\!]_{\mathcal{P}}$ is defined inductively over the structure of t by

$$[\![t]\!]_{\mathcal{P}}(m) = \begin{cases} 1 & \text{if } t = 0 \text{ and } m = 0 \\ 1 & \text{if } t = x \text{ and } m = 1_x \\ \displaystyle\sum_{\substack{m_1,m_2\in\mathcal{M} \text{ with} \\ \forall x\in\mathcal{V}.\, m(x)=m_1(x)+m_2(x)}} [\![t_1]\!]_{\mathcal{P}}(m_1)\cdot[\![t_2]\!]_{\mathcal{P}}(m_2) & \text{if } t = t_1 \parallel t_2 \\ \displaystyle\sum_{i=1}^{n} q_i [\![t_i]\!]_{\mathcal{P}}((1/\lambda)\cdot m) & \text{if } t = a.\bigoplus_{i=1}^{n}[q_i]t_i \\ 0 & \text{otherwise} \end{cases}$$

for all $m \in \mathcal{M}$, with $((1/\lambda)\cdot m)$ being the multiplicity defined by $((1/\lambda)\cdot m)(x) = (1/\lambda)\cdot m(x)$. Notice that $[\![t]\!]_{\mathcal{P}} = \delta([\![t]\!]_{\mathcal{M}})$ for all deterministic process terms $t \in \mathbb{T}_{\text{det}}(\Sigma_{\text{PA}})$.

For important probabilistic multiplicities we use the same symbols as for multiplicities but it will be always clear from the context if we refer to probabilistic multiplicities or multiplicities. By $0 \in \mathcal{P}$ we mean the probabilistic multiplicity that gives probability 1 to the multiplicity $0 \in \mathcal{M}$. By $n_V \in \mathcal{P}$ we mean the probabilistic multiplicity that gives probability 1 to the multiplicity $n_V \in \mathcal{M}$.

Definition 5.13. The *probabilistic distance approximation from above* $\mathbf{P}\colon (\mathcal{P}\times\mathcal{E}) \to [0,\lambda]\cup\{1\}$ is defined by

$$\mathbf{P}(p,e) = \sum_{m\in\mathcal{M}} p(m)\cdot\mathbf{D}(m,e)$$

for all $p \in \mathcal{P}$ and $e \in \mathcal{E}$.

Example 5.14. Consider the probabilistic process term $t = a.([0.5](x \parallel x) \oplus [0.5]0)$ and the substitutions $\sigma_1(x) = a.a.0$ and $\sigma_2(x) = a.([0.9]a.0 \oplus [0.1]0)$ with $\mathbf{d}(\sigma_1(x),\sigma_2(x)) = 0.1\lambda$. Then $\mathbf{d}(\sigma_1(t),\sigma_2(t)) = 0.5\lambda^2(1-(1-0.1)^2)$. The probabilistic multiplicity of

t is $[\![t]\!]_{\mathscr{P}}((2\lambda)_x) = 0.5$ and $[\![t]\!]_{\mathscr{P}}(0) = 0.5$. Then, $\mathbf{D}((2\lambda)_x, \mathbf{d}(\sigma_1,\sigma_2)) = \lambda(1-(1-0.1)^{2\lambda})$ and $\mathbf{D}(0,\mathbf{d}(\sigma_1,\sigma_2)) = 0$. Hence, we get the probabilistic distance approximation $\mathbf{P}([\![t]\!]_{\mathscr{P}}, \mathbf{d}(\sigma_1,\sigma_2)) = 0.5\lambda(1-(1-0.1)^{2\lambda}) \geq 0.5\lambda^2(1-(1-0.1)^2) = \mathbf{d}(\sigma_1(t),\sigma_2(t))$.

Remark 5.15. The functional $\mathbf{P}$ shows a very important interaction between probabilistic choice and process replication. Consider the process term $t = a.([q](x \parallel x) \oplus [1-q]0)$ with $q \in (0,1)$, and any closed substitutions σ_1, σ_2 with $\mathbf{d}(\sigma_1(x),\sigma_2(x)) = \epsilon$ for any $\epsilon \in [0,1)$. In the probabilistic distance approximation $\mathbf{P}([\![t]\!]_{\mathscr{P}}, \mathbf{d}(\sigma_1,\sigma_2))$ the deterministic distance approximation $\mathbf{D}((2\lambda)_x, \mathbf{d}(\sigma_1,\sigma_2)) = \lambda(1-(1-\epsilon/\lambda)^{2\lambda})$ of the synchronous parallel execution $x \parallel x$ of two instances of x is weighted by the likelihood q of its realization. Hence, $\mathbf{P}([\![t]\!]_{\mathscr{P}}, \mathbf{d}(\sigma_1,\sigma_2)) = q\lambda(1-(1-\epsilon/\lambda)^{2\lambda})$. From Bernoulli's inequality $\frac{1}{m}\lambda(1-(1-\epsilon/\lambda)^n) \leq \epsilon$ if $m \geq n$, we get $q\lambda(1-(1-\epsilon/\lambda)^{2\lambda}) \leq 2q\lambda\epsilon$. Hence, the distance between instances of two copies running synchronously in parallel with a probability of $q = 0.5/\lambda$ is at most the distance between those instances running (non-replicated) with a probability of 1.0.

Notice that for all deterministic terms $t \in \mathbb{T}_{\mathrm{det}}(\Sigma_{\mathrm{PA}})$, from $[\![t]\!]_{\mathscr{P}} = \delta([\![t]\!]_{\mathscr{M}})$ it follows $\mathbf{P}([\![t]\!]_{\mathscr{P}}, \mathbf{d}(\sigma_1,\sigma_2)) = \mathbf{D}([\![t]\!]_{\mathscr{M}}, \mathbf{d}(\sigma_1,\sigma_2))$.

The functional $\mathbf{P}$ defines an upper bound on the bisimulation distance of probabilistic processes.

Proposition 5.16. *Let $t \in \mathbb{T}_{prob}(\Sigma_{PA})$ be a probabilistic process term and σ_1, σ_2 be closed substitutions. Then $\mathbf{d}(\sigma_1(t),\sigma_2(t)) \leq \mathbf{P}([\![t]\!]_{\mathscr{P}}, \mathbf{d}(\sigma_1,\sigma_2))$.*

Proof. It is easy to verify that the denotations defined for $t \in \mathbb{T}_{\mathrm{prob}}(\Sigma_{\mathrm{PA}})$ are exactly those that are computed by the least fixed point of $\mathbf{F}$ in Section 5.3. The Proposition follows then from Theorem 5.45 and the observation that $\mathbf{P}([\![t]\!]_{\mathscr{P}}, \mathbf{d}(\sigma_1,\sigma_2)) = \mathbf{A}([\![t]\!], \mathbf{d}(\sigma_1,\sigma_2))$. □

We conclude this section by introducing an ordering relation over $\mathscr{P}$ based on the ordering over $\mathscr{M}$ (Definition 5.8) such that the probabilistic distance approximation $\mathbf{P}$ is monotone in both arguments.

Definition 5.17. We order the elements of $\mathscr{P}$ by $\sqsubseteq \subseteq \mathscr{P} \times \mathscr{P}$ defined as

$$p_1 \sqsubseteq p_2 \text{ iff there is a } \omega \in \Omega(p_1,p_2) \text{ with } m_1 \sqsubseteq m_2 \text{ if } \omega(m_1,m_2) > 0 \text{ for all } m_1, m_2 \in \mathscr{M}$$

for all $p_1, p_2 \in \mathscr{P}$.

It is clear that $\mathbf{P}(p,e) = 0$ if $p = 0 \in \mathscr{P}$ or $e = 0 \in \mathscr{E}$.

Proposition 5.18. *$(\mathscr{P}, \sqsubseteq)$ is a cpo with least element* 0.

Proof. Reflexivity is immediate since for each $p \in \mathscr{P}$ we get $p \sqsubseteq p$ by the matching $\omega \in \Omega(p,p)$ defined by $\omega(m,m) = p(m)$ for all $m \in \mathscr{M}$.

To show transitivity assume $p_1, p_2, p_3 \in \mathscr{P}$ with $p_1 \sqsubseteq p_2$ and $p_2 \sqsubseteq p_3$, namely there are matchings $\omega_1 \in \Omega(p_1,p_2)$ and $\omega_2 \in \Omega(p_2,p_3)$ such that $m_1 \sqsubseteq m_2$ for all $m_1, m_2 \in \mathscr{M}$ with $\omega_1(m_1,m_2) > 0$, and $m_2 \sqsubseteq m_3$ for all $m_2, m_3 \in \mathscr{M}$ with $\omega_2(m_2,m_3) > 0$. Define the distribution $\omega \in \Delta(\mathscr{M} \times \mathscr{M})$ as:

$$\omega(m_1,m_3) = \sum_{\substack{m_2 \in \mathscr{M} \\ p_2(m_2) > 0}} \frac{\omega_1(m_1,m_2) \cdot \omega_2(m_2,m_3)}{p_2(m_2)}$$

First we show that ω is a matching $\omega \in \Omega(p_1, p_3)$ for p_1 and p_3. We have

$$\begin{aligned}
&\sum_{m_3 \in \mathscr{M}} \omega(m_1, m_3) \\
=&\sum_{m_3 \in \mathscr{M}} \sum_{\substack{m_2 \in \mathscr{M} \\ p_2(m_2) > 0}} \frac{\omega_1(m_1, m_2) \cdot \omega_2(m_2, m_3)}{p_2(m_2)} \\
=&\sum_{\substack{m_2 \in \mathscr{M} \\ p_2(m_2) > 0}} \left(\omega_1(m_1, m_2) \cdot \frac{1}{p_2(m_2)} \sum_{m_3 \in \mathscr{M}} \omega_2(m_2, m_3) \right) \\
=&\sum_{\substack{m_2 \in \mathscr{M} \\ p_2(m_2) > 0}} \omega_1(m_1, m_2) \\
=&p_1(m_1)
\end{aligned}$$

and, analogously, $\sum_{m_1 \in \mathscr{M}} \omega(m_1, m_3) = p_3(m_3)$, thus confirming that $\omega \in \Omega(p_1, p_3)$. It remains to show that $\omega(m_1, m_3) > 0$ implies $m_1 \sqsubseteq m_3$. If $\omega(m_1, m_3) > 0$ then there exists at least one $m_2 \in \mathscr{M}$ with $\omega_1(m_1, m_2), \omega_2(m_2, m_3) > 0$, thus implying $m_1 \sqsubseteq m_2 \sqsubseteq m_3$ and $m_1 \sqsubseteq m_3$ from the transitivity of the ordering $\sqsubseteq$ over $\mathscr{M}$.

To show antisymmetry assume $p_1, p_2 \in \mathscr{P}$ with $p_1 \sqsubseteq p_2$ and $p_2 \sqsubseteq p_1$. Relation $p_1 = p_2$ follows from the observation that the composition of the respective transportation schedules are the diagonal/identity transportation plans. In detail, it is not hard to see that the only matching $\omega \in \Omega(p_1, p_1)$ satisfying $m \sqsubseteq m'$ for all $m, m' \in \mathscr{M}$ with $\omega(m, m') > 0$ is the matching with $\omega(m, m) = p_1(m)$. From $p_1 \sqsubseteq p_2$ and $p_2 \sqsubseteq p_1$ there are matchings $\omega_1 \in \Omega(p_1, p_2)$ and $\omega_2 \in \Omega(p_2, p_1)$ such that $m_1 \sqsubseteq m_2$ for all $m_1, m_2 \in \mathscr{M}$ with $\omega_1(m_1, m_2) > 0$ and $m_2 \sqsubseteq m_1$ for all $m_1, m_2 \in \mathscr{M}$ with $\omega_2(m_2, m_1) > 0$. Define the distribution $\omega \in \Delta(\mathscr{M} \times \mathscr{M})$ as:

$$\omega(m_1, m_1') = \sum_{\substack{m_2 \in \mathscr{M} \\ p_2(m_2) > 0}} \frac{\omega_1(m_1, m_2) \cdot \omega_2(m_2, m_1')}{p_2(m_2)}$$

By following the proof of transitivity (with p_3 replaced by p_1), we get that ω is a matching $\omega \in \Omega(p_1, p_1)$ such that $m_1 \sqsubseteq m_1'$ for all $m_1, m_1' \in \mathscr{M}$ with $\omega(m_1, m_1') > 0$, thus implying $m_1 \sqsubseteq m_2 \sqsubseteq m_1'$ for all m_1, m_2, m_1' with $\omega_1(m_1, m_2), \omega_2(m_2, m_1') > 0$. Since $m_1 \sqsubseteq m_1'$ for all $m_1, m_1' \in \mathscr{M}$ with $\omega(m_1, m_1') > 0$, we get $\omega(m, m) = p_1(m)$ for all $m \in \mathscr{M}$. It follows that for all m_1, m_2, m_1' with $\omega_1(m_1, m_2), \omega_2(m_2, m_1') > 0$ we have $m_1 = m_2 = m_1'$. Summarizing, $p_1(m) = \omega(m, m) = (\omega_1(m, m) \cdot \omega_2(m, m))/p_2(m) = (p_1(m) \cdot p_1(m))/p_2(m)$, thus implying $p_1 = p_2$.

Now we show that $(\mathscr{P}, \sqsubseteq)$ is a cpo. Assume a countable ascending chain of probabilistic multiplicities $p_1 \sqsubseteq p_2 \sqsubseteq \ldots$, with, for each $k \in \mathbb{N}$, $\omega_{(k,k+1)} \in \Omega(p_k, p_{k+1})$ a matching for p_k and p_{k+1} witnessing $p_k \sqsubseteq p_{k+1}$. For each $k \in \mathbb{N}$, we will define a probabilistic multiplicity p_k' and a matching $\omega_k \in \Omega(p_k, p_k')$ for p_k and p_k' witnessing $p_k \sqsubseteq p_k'$. Then, we prove that all p_k' coincide, namely there exists a $p \in \mathscr{P}$ with $p = p_1' = p_2' \ldots$. Hence $p_k \sqsubseteq p$ for all $k \in \mathbb{N}$. Finally, we show that $p \sqsubseteq p'$ for all p' satisfying $p_k \sqsubseteq p'$ for all $k \in \mathbb{N}$, thus concluding that $p = \sup\{p_1, p_2, \ldots\}$.

We start by defining for all $k < k' \in \mathbb{N}$ the function $\omega_{(k,k')} \colon \mathscr{M} \times \mathscr{M} \to [0,1]$ by

$$\omega_{(k,k')}(m_k, m_{k'}) = \sum_{\substack{m_h \in \mathscr{M} \\ k+1 \le h \le k'-1 \\ \omega_{(h,h+1)}(m_h, m_{h+1}) > 0}} \frac{\prod_{h=k}^{k'-1} \omega_{(h,h+1)}(m_h, m_{h+1})}{\prod_{h=k+1}^{k'-1} p_h(m_h)}$$

for $m_k, m_{k'} \in \mathscr{M}$. By following the proof for transitivity property, we infer that $\omega_{(k,k')}$ is a matching $\omega_{(k,k')} \in \Omega(p_k, p_{k'})$ for p_k and $p_{k'}$ witnessing $p_k \sqsubseteq p_{k'}$.

For each $k \in \mathbb{N}$ we first define the function $\omega_k \colon \mathscr{M} \times \mathscr{M} \to [0,1]$ by

$$\omega_k(m_k, m) = \lim_{k' \to \infty} \sum_{\substack{m_{k'} \in \mathscr{M} \\ \forall k' \le h. \exists m_h \in \mathscr{M}. \\ \forall h \ge k'. \omega_{(h,h+1)}(m_h, m_{h+1}) > 0 \\ m = \sup\{m_h \mid h \ge k'\}}} \omega_{(k,k')}(m_k, m_{k'})$$

and, then, the function $p'_k \colon \mathscr{M} \to [0,1]$ by

$$p'_k(m) = \sum_{m_k \in \mathscr{M}} \omega_k(m_k, m).$$

Now we prove that $\omega_k(m_k, m)$ is well-defined, p'_k is a distribution and ω_k is a matching for p_k and p'_k. We start with showing that $\omega_k(m_k, m)$ is well-defined, namely the limit used in the definition exists. It is enough to show that for each $m_k \in \mathscr{M}$, the sequence

$$\left(\sum_{\substack{m_{k'} \in \mathscr{M} \\ \forall k' \le h. \exists m_h \in \mathscr{M}. \\ \forall h \ge k'. \omega_{(h,h+1)}(m_h, m_{h+1}) > 0 \\ m = \sup\{m_h \mid h \ge k'\}}} \omega_{(k,k')}(m_k, m_{k'}) \right)_{k' \in \mathbb{N}_{>k}}$$

is descending. In fact, since this sequence is bounded (all elements are in $[0,1]$), if we prove that it is descending then we infer that it converges. Descending property follows from

$$\begin{aligned}
& \sum_{\substack{m_{k'+1} \in \mathscr{M} \\ \forall k'+1 \le h. \exists m_h \in \mathscr{M}. \\ \forall h \ge k'+1. \omega_{(h,h+1)}(m_h, m_{h+1}) > 0 \\ m = \sup\{m_h \mid h \ge k'+1\}}} \omega_{(k,k'+1)}(m_k, m_{k'+1}) \\
= & \sum_{\substack{m_{k'+1} \in \mathscr{M} \\ \forall k'+1 \le h. \exists m_h \in \mathscr{M}. \\ \forall h \ge k'+1. \omega_{(h,h+1)}(m_h, m_{h+1}) > 0 \\ m = \sup\{m_h \mid h \ge k'+1\}}} \sum_{\substack{m_{k'} \in \mathscr{M} \\ p_{k'}(m_{k'}) > 0}} \frac{\omega_{(k,k')}(m_k, m_{k'}) \cdot \omega_{(k',k'+1)}(m_{k'}, m_{k'+1})}{p_{k'}(m_{k'})} \\
= & \sum_{\substack{m_{k'+1} \in \mathscr{M} \\ \forall k'+1 \le h. \exists m_h \in \mathscr{M}. \\ \forall h \ge k'+1. \omega_{(h,h+1)}(m_h, m_{h+1}) > 0 \\ m = \sup\{m_h \mid h \ge k'+1\}}} \sum_{\substack{m_{k'} \in \mathscr{M} \\ \forall k' \le h. \exists m_h \in \mathscr{M}. \\ \forall h \ge k'+1. \omega_{(h,h+1)}(m_h, m_{h+1}) > 0 \\ m = \sup\{m_h \mid h \ge k'\}}} \frac{\omega_{(k,k')}(m_k, m_{k'}) \cdot \omega_{(k',k'+1)}(m_{k'}, m_{k'+1})}{p_{k'}(m_{k'})} \\
\le & \sum_{\substack{m_{k'} \in \mathscr{M} \\ \forall k' \le h. \exists m_h \in \mathscr{M}. \\ \forall h \ge k'. \omega_{(h,h+1)}(m_h, m_{h+1}) > 0 \\ m = \sup\{m_h \mid h \ge k'\}}} \omega_{(k,k')}(m_k, m_{k'}) \cdot \sum_{\substack{m_{k'+1} \in \mathscr{M} \\ \forall k'+1 \le h. \exists m_h \in \mathscr{M}. \\ \forall h \ge k'+1. \omega_{(h,h+1)}(m_h, m_{h+1}) > 0 \\ m = \sup\{m_h \mid h \ge k'+1\}}} \frac{\omega_{(k',k'+1)}(m_{k'}, m_{k'+1})}{p_{k'}(m_{k'})}
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{\substack{m_{k'} \in \mathscr{M} \\ \forall k' \leq h. \exists m_h \in \mathscr{M}. \\ \forall h \geq k'. \omega_{(h,h+1)}(m_h, m_{h+1}) > 0 \\ m = \sup\{m_h \mid h \geq k'\}}} \omega_{(k,k')}(m_k, m_{k'}) \cdot \sum_{m_{k'+1} \in \mathscr{M}} \frac{\omega_{(k',k'+1)}(m_{k'}, m_{k'+1})}{p_{k'}(m_{k'})} \\
&= \sum_{\substack{m_{k'} \in \mathscr{M} \\ \forall k' \leq h. \exists m_h \in \mathscr{M}. \\ \forall h \geq k'. \omega_{(h,h+1)}(m_h, m_{h+1}) > 0 \\ m = \sup\{m_h \mid h \geq k'\}}} \omega_{(k,k')}(m_k, m_{k'}) \cdot \frac{p_{k'}(m_{k'})}{p_{k'}(m_{k'})} \\
&= \sum_{\substack{m_{k'} \in \mathscr{M} \\ \forall k' \leq h. \exists m_h \in \mathscr{M}. \\ \forall h \geq k'. \omega_{(h,h+1)}(m_h, m_{h+1}) > 0 \\ m = \sup\{m_h \mid h \geq k'\}}} \omega_{(k,k')}(m_k, m_{k'})
\end{aligned}
$$

for all $k' \in \mathbb{N}_{>k}$.

Then we prove that p'_k is a distribution. Property $p'_k(m) \geq 0$ for all $m \in \mathscr{M}$ is immediate. It remains to prove $\sum_{m \in \mathscr{M}} p'_k(m) = 1$. We have

$$
\begin{aligned}
&\sum_{m \in \mathscr{M}} p'_k(m) \\
&= \sum_{m, m_k \in \mathscr{M}} \omega_k(m_k, m) \\
&= \sum_{m, m_k \in \mathscr{M}} \lim_{k' \to \infty} \sum_{\substack{m_{k'} \in \mathscr{M} \\ \forall k' \leq h. \exists m_h \in \mathscr{M}. \\ \forall h \geq k'. \omega_{(h,h+1)}(m_h, m_{h+1}) > 0 \\ m = \sup\{m_h \mid h \geq k'\}}} \omega_{(k,k')}(m_k, m_{k'}) \\
&= \lim_{k' \to \infty} \sum_{m, m_k \in \mathscr{M}} \sum_{\substack{m_{k'} \in \mathscr{M} \\ \forall k' \leq h. \exists m_h \in \mathscr{M}. \\ \forall h \geq k'. \omega_{(h,h+1)}(m_h, m_{h+1}) > 0 \\ m = \sup\{m_h \mid h \geq k'\}}} \omega_{(k,k')}(m_k, m_{k'}) \\
&= \lim_{k' \to \infty} \sum_{m_k \in \mathscr{M}} \sum_{\substack{m_{k'} \in \mathscr{M} \\ \forall k' \leq h. \exists m_h \in \mathscr{M}. \\ \forall h \geq k'. \omega_{(h,h+1)}(m_h, m_{h+1}) > 0}} \omega_{(k,k')}(m_k, m_{k'}) \\
&= \lim_{k' \to \infty} \sum_{m_k \in \mathscr{M}} \sum_{m_{k'} \in \mathscr{M}} \omega_{(k,k')}(m_k, m_{k'}) \\
&= \lim_{k' \to \infty} \sum_{m_k \in \mathscr{M}} p_k(m_k) \\
&= \lim_{k' \to \infty} 1 \\
&= 1.
\end{aligned}
$$

Now we prove that ω_k is a matching $\omega_k \in \Omega(p_k, p'_k)$ for p_k and p'_k witnessing $p_k \sqsubseteq p'_k$. The marginal distribution of ω_k with respect to p_k is such that for all $m_k \in \mathscr{M}$ we have

$$\sum_{m \in \mathscr{M}} \omega_k(m_k, m)$$

$$
\begin{aligned}
&= \sum_{m\in\mathscr{M}} \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\le h.\exists m_h\in\mathscr{M}.\\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0\\ m=\sup\{m_h\mid h\ge k'\}}} \omega_{(k,k')}(m_k,m_{k'})\\
&= \lim_{k'\to\infty} \sum_{m\in\mathscr{M}} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\le h.\exists m_h\in\mathscr{M}.\\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0\\ m=\sup\{m_h\mid h\ge k'\}}} \omega_{(k,k')}(m_k,m_{k'})\\
&= \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\le h.\exists m_h\in\mathscr{M}.\\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0}} \omega_{(k,k')}(m_k,m_{k'})\\
&= \lim_{k'\to\infty} \sum_{m_{k'}\in\mathscr{M}} \omega_{(k,k')}(m_k,m_{k'})\\
&= \lim_{k'\to\infty} p_k(m_k)\\
&= p_k(m_k)
\end{aligned}
$$

and the marginal distribution of ω_k with respect to p'_k is, by definition, $\sum_{m_k\in\mathscr{M}} \omega_k(m_k,m) = p'_k(m)$, thus confirming that $\omega_k \in \Omega(p_k,p'_k)$.

Summarizing, each p'_k is a probabilistic distribution $p'_k \in \mathscr{P}$ with $p_k \sqsubseteq p'_k$. Now, for all $k\in\mathbb{N}$ we get

$$
\begin{aligned}
&p'_k(m)\\
&= \sum_{m_k\in\mathscr{M}} \omega_k(m_k,m)\\
&= \sum_{m_k\in\mathscr{M}} \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\le h.\exists m_h\in\mathscr{M}.\\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0\\ m=\sup\{m_h\mid h\ge k'\}}} \omega_{(k,k')}(m_k,m_{k'})\\
&= \lim_{k'\to\infty} \sum_{m_k\in\mathscr{M}} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\le h.\exists m_h\in\mathscr{M}.\\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0\\ m=\sup\{m_h\mid h\ge k'\}}} \omega_{(k,k')}(m_k,m_{k'})\\
&= \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\le h.\exists m_h\in\mathscr{M}.\\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0\\ m=\sup\{m_h\mid h\ge k'\}}} \sum_{m_k\in\mathscr{M}} \omega_{(k,k')}(m_k,m_{k'})\\
&= \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\le h.\exists m_h\in\mathscr{M}.\\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0\\ m=\sup\{m_h\mid h\ge k'\}}} p_{k'}(m_{k'})
\end{aligned}
$$

thus implying that $p'_k(m)$ does not depend on the index k, namely all p'_k with $k\in\mathbb{N}$ are the same distribution.

Let us name p the probabilistic multiplicity $p = p'_1, p'_2, \ldots$. Since $p_k \sqsubseteq p$ for each $k\in\mathbb{N}$, to prove that $p = \sup\{p_1,p_2,\ldots\}$ we have to show that $p \sqsubseteq p'$ for each $p'\in\mathscr{P}$ satisfying

$p_k \sqsubseteq p'$ for each $k \in \mathbb{N}$. Given such a p', for each $k \in \mathbb{N}$ let $\omega'_k \in \Omega(p_k, p')$ be a matching witnessing $p_k \sqsubseteq p'$. We have to provide a matching for p and p'. Define the function $\omega \colon \mathscr{M} \times \mathscr{M} \to [0,1]$ by

$$\omega(m,m') = \lim_{k\to\infty} \sum_{\substack{m_k \in \mathscr{M} \\ \forall k<h.\exists m_h \in \mathscr{M}. \\ \forall h \geq k.\omega_{(h,h+1)}(m_h,m_{h+1})>0 \\ m=\sup\{m_h \mid h\geq 1\}}} \omega'_k(m_k,m')$$

for all $m, m' \in \mathscr{M}$. We prove that ω is well-defined and, then, that it is a matching $\omega \in \Omega(p,p')$ for p and p'. First we observe that the sequence

$$\left(\sum_{\substack{m_k \in \mathscr{M} \\ \forall k<h.\exists m_h \in \mathscr{M}. \\ \forall h \geq k.\omega_{(h,h+1)}(m_h,m_{h+1})>0 \\ m=\sup\{m_h \mid h\geq 1\}}} \omega'_k(m_k,m') \right)_{k\in\mathbb{N}}$$

is descending since for $k < k' \in \mathbb{N}$ we have that for all $m_h \in \mathscr{M}$ with $k < h$ such that $\omega_{(h,h+1)}(m_h, m_{h+1}) > 0$, whenever $m = \sup\{m_h \mid h \geq k'\}$ then it holds $m = \sup\{m_h \mid h \geq k\}$. Then we observe that this sequence is bounded (all elements are in $[0,1]$) and we conclude that it converges. Therefore ω is well-defined. Now we have

$$\begin{aligned}
&\sum_{m\in\mathscr{M}} \omega(m,m') \\
&= \sum_{m\in\mathscr{M}} \lim_{k\to\infty} \sum_{\substack{m_k \in \mathscr{M} \\ \forall k<h.\exists m_h \in \mathscr{M}. \\ \forall h \geq k.\omega_{(h,h+1)}(m_h,m_{h+1})>0 \\ m=\sup\{m_h \mid h\geq 1\}}} \omega'_k(m_k,m') \\
&= \lim_{k\to\infty} \sum_{m\in\mathscr{M}} \sum_{\substack{m_k \in \mathscr{M} \\ \forall k<h.\exists m_h \in \mathscr{M}. \\ \forall h \geq k.\omega_{(h,h+1)}(m_h,m_{h+1})>0 \\ m=\sup\{m_h \mid h\geq 1\}}} \omega'_k(m_k,m') \\
&= \lim_{k\to\infty} \sum_{m_k\in\mathscr{M}} \omega'_k(m_k,m') \\
&= \lim_{k\to\infty} p'(m') \\
&= p'(m')
\end{aligned}$$

and

$$\begin{aligned}
&\sum_{m'\in\mathscr{M}} \omega(m,m') \\
&= \sum_{m'\in\mathscr{M}} \lim_{k\to\infty} \sum_{\substack{m_k \in \mathscr{M} \\ \forall k<h.\exists m_h \in \mathscr{M}. \\ \forall h \geq k.\omega_{(h,h+1)}(m_h,m_{h+1})>0 \\ m=\sup\{m_h \mid h\geq 1\}}} \omega'_k(m_k,m') \\
&= \lim_{k\to\infty} \sum_{m'\in\mathscr{M}} \sum_{\substack{m_k \in \mathscr{M} \\ \forall k<h.\exists m_h \in \mathscr{M}. \\ \forall h \geq k.\omega_{(h,h+1)}(m_h,m_{h+1})>0 \\ m=\sup\{m_h \mid h\geq 1\}}} \omega'_k(m_k,m')
\end{aligned}$$

$$
\begin{aligned}
&= \lim_{k\to\infty} \sum_{\substack{m_k\in\mathscr{M}\\ \forall k<h.\exists m_h\in\mathscr{M}.\\ \forall h\geq k.\omega_{(h,h+1)}(m_h,m_{h+1})>0\\ m=\sup\{m_h\mid h\geq 1\}}} \sum_{m'\in\mathscr{M}} \omega'_k(m_k,m') \\
&= \lim_{k\to\infty} \sum_{\substack{m_k\in\mathscr{M}\\ \forall k<h.\exists m_h\in\mathscr{M}.\\ \forall h\geq k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0\\ m=\sup\{m_h\mid h\geq 1\}}} p_k(m_k) \\
&= p(m)
\end{aligned}
$$

thus confirming that ω is a matching for p and p' and

$$
\sup(p_k)_{k\in\mathbb{N}}(m) = \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\leq h.\exists m_h\in\mathscr{M}.\\ \forall h\geq k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0\\ m=\sup\{m_h\mid h\geq k'\}}} p_{k'}(m_{k'}) \tag{5.1}
$$

for all $m\in\mathscr{M}$.

We conclude by observing that it is clear that $0\in\mathscr{P}$ is the least element of $(\mathscr{P},\sqsubseteq)$. □

Proposition 5.19. *Let $p,p'\in\mathscr{P}$ and $e,e'\in\mathscr{E}$. Then*

1. $\mathbf{P}(p,e)\leq\mathbf{P}(p',e)$ *if* $p\sqsubseteq p'$;
2. $\mathbf{P}(p,e)\leq\mathbf{P}(p,e')$ *if* $e\sqsubseteq e'$.

Proof. To prove monotonicity of $\mathbf{P}$ we exploit monotonicity of $\mathbf{D}$ (Proposition 5.11). Consider first case 1. Let $\omega\in\Omega(p,p')$ be a matching for p and p' witnessing $p\sqsubseteq p'$. We have

$$
\begin{aligned}
&\mathbf{P}(p,e) \\
&= \sum_{m\in\mathscr{M}} p(m)\,\mathbf{D}(m,e) \\
&= \sum_{m\in\mathscr{M}}\sum_{m'\in\mathscr{M}} \omega(m,m')\,\mathbf{D}(m,e) \\
&\leq \sum_{m\in\mathscr{M}}\sum_{m'\in\mathscr{M}} \omega(m,m')\,\mathbf{D}(m',e) && \text{(by Proposition 5.11.1)} \\
&= \sum_{m'\in\mathscr{M}}\sum_{m\in\mathscr{M}} \omega(m,m')\,\mathbf{D}(m',e) \\
&= \sum_{m'\in\mathscr{M}} p'(m')\,\mathbf{D}(m',e) \\
&= \mathbf{P}(p',e).
\end{aligned}
$$

Consider now case 2. We have

$$
\begin{aligned}
&\mathbf{P}(p,e) \\
&= \sum_{m\in\mathscr{M}} p(m)\cdot\mathbf{D}(m,e)
\end{aligned}
$$

$$\leq \sum_{m \in \mathscr{M}} p(m) \cdot \mathbf{D}(m, e') \qquad \text{(by Proposition 5.11.2)}$$
$$= \mathbf{P}(p, e').$$

□

5.2.3 Denotation of nondeterministic probabilistic process terms

The denotation of a nondeterministic probabilistic process term $t \in \mathbb{T}(\Sigma_{\mathrm{PA}})$ is a set of probabilistic multiplicities $P \subseteq \mathscr{P}$ that describes by $p \in P$ some resolution of the nondeterministic choices in t such that the process evolves as a probabilistic process described by p. We construct a Hoare powerdomain over the probabilistic multiplicities $\mathscr{P}$ and use as canonical representation for any set of probabilistic multiplicities $P \subseteq \mathscr{P}$ the downward closure defined as $\downarrow P = \{p \in \mathscr{P} \mid p \sqsubseteq p' \text{ for some } p' \in P\}$. We use downward closed sets such that $\mathscr{D}$ will form a cpo with the order defined below (Definition 5.24) (esp. satisfies antisymmetry, cf. Proposition 5.25).

Definition 5.20. A *nondeterministic probabilistic multiplicity* is a downward closed subset of the set probabilistic multiplicities $\mathscr{P}$. The set of all nondetermnistic probabilistic multiplicites is denoted by $\mathscr{D}$.

The denotation of term t is the nondetermnistic probabilistic multiplicity $[\![t]\!]$ defined inductively over the structure of t by

$$[\![t]\!] = \downarrow \begin{cases} \{[\![t]\!]_{\mathscr{P}}\} & \text{if } t = 0 \text{ or } t = x \\ \left\{ p \in \mathscr{P} \;\middle|\; \begin{array}{l} \exists p_1 \in [\![t_1]\!].\exists p_2 \in [\![t_2]\!].\forall m \in \mathscr{M}. \\ (p(m) = \sum\limits_{\substack{m_1, m_2 \in \mathscr{M} \text{ with} \\ \forall x \in \mathscr{V}. m(x) = m_1(x) + m_2(x)}} p_1(m_1) \cdot p_2(m_2)) \end{array} \right\} & \text{if } t = t_1 \parallel t_2 \\ \left\{ p' \in \mathscr{P} \mid \exists p_1 \in [\![t_1]\!] \ldots \exists p_n \in [\![t_n]\!].p' = \sum\limits_{i=1}^{n} q_i \cdot (\lambda \cdot p_i) \right\} & \text{if } t = a.\bigoplus_{i=1}^{n} [q_i] t_i \\ [\![t_1]\!] \cup [\![t_2]\!] & \text{if } t = t_1 + t_2 \end{cases}$$

with $(\lambda \cdot p_i)$ being the probabilistic multiplicity defined by $(\lambda \cdot p_i)(\lambda \cdot m) = p_i(m)$ for all $m \in \mathscr{M}$.

Notice that $[\![t]\!] = \downarrow \{[\![t]\!]_{\mathscr{P}}\}$ for all probabilistic process terms $t \in \mathbb{T}_{\mathrm{prob}}(\Sigma_{\mathrm{PA}})$.

By $0 \in \mathscr{D}$ we mean the singleton set containing the probabilistic multiplicity $0 \in \mathscr{P}$, and by $n_V \in \mathscr{D}$ the downward closure of the singleton set with element $n_V \in \mathscr{P}$.

Definition 5.21. The *nondeterministic probabilistic distance approximation from above* $\mathbf{A}\colon (\mathscr{D} \times \mathscr{E}) \to [0, \lambda] \cup \{1\}$ is defined by

$$\mathbf{A}(P, e) = \sup_{p \in P} \mathbf{P}(p, e)$$

for all $P \in \mathscr{D}$ and $e \in \mathscr{E}$.

Example 5.22. Consider the nondeterministic probabilistic process term $t = a.([0.5](x \parallel x) \oplus [0.5]0) + y$, and the substitutions $\sigma_1(x) = a.a.0$, $\sigma_2(x) = a.([0.9]a.0 \oplus [0.1]0)$, $\sigma_1(y) = b.b.0$ and $\sigma_2(y) = b.([0.8]b.0 \oplus [0.2]0)$. It is clear that $\mathbf{d}(\sigma_1(x), \sigma_2(x)) = 0.1\lambda$ and $\mathbf{d}(\sigma_1(y), \sigma_2(y)) = 0.2\lambda$. Now, $\mathbf{d}(\sigma_1(t), \sigma_2(t)) = \max\{0.5\lambda^2(1-(1-0.1)^2), 0.2\lambda\}$. The nondeterministic probabilistic multiplicity of t is $[\![t]\!] = \downarrow\{p_1, p_2\}$, with p_1 and p_2 defined by $p_1((2\lambda)_x) = 0.5$, $p_1(0) = 0.5$ and $p_2(1_y) = 1.0$. Thus $\mathbf{A}([\![t]\!], \mathbf{d}(\sigma_1, \sigma_2)) = \max(\mathbf{P}(p_1, \mathbf{d}(\sigma_1, \sigma_2)), \mathbf{P}(p_2, \mathbf{d}(\sigma_1, \sigma_2))) = \max(0.5\lambda(1-(1-0.1)^{2\lambda}), 0.2\lambda)$. We conclude that $\mathbf{A}([\![t]\!], \mathbf{d}(\sigma_1, \sigma_2)) \geq \mathbf{d}(\sigma_1(t), \sigma_2(t))$.

Notice that for all probabilistic process terms $t \in \mathbb{T}_{\text{prob}}(\Sigma_{\text{PA}})$, from $[\![t]\!] = \downarrow\{[\![t]\!]_{\mathscr{P}}\}$ and from $\mathbf{A}(P, e) = \mathbf{A}(\downarrow P, e)$ for any $P \subseteq \mathscr{P}$ and $e \in \mathscr{E}$, we infer $\mathbf{A}([\![t]\!], \mathbf{d}(\sigma_1, \sigma_2)) = \mathbf{P}([\![t]\!]_{\mathscr{P}}, \mathbf{d}(\sigma_1, \sigma_2))$.

The functional $\mathbf{A}$ defines an upper bound on the bisimulation distance of nondeterministic probabilistic process terms.

Theorem 5.23. *Let $t \in \mathbb{T}(\Sigma_{PA})$ be a nondeterministic probabilistic process term and σ_1, σ_2 be closed substitutions. Then $\mathbf{d}(\sigma_1(t), \sigma_2(t)) \leq \mathbf{A}([\![t]\!], \mathbf{d}(\sigma_1, \sigma_2))$.*

Proof. It is easy to verify that the denotations defined for $t \in \mathbb{T}(\Sigma_{\text{PA}})$ are exactly those that are computed by the least fixed point of $\mathbf{F}$ in Section 5.3. The Proposition follows then from Theorem 5.45. □

Theorem 5.23 shows that the denotation of a process term is adequate to define an upper bound to the distance between the closed instances of that process term. The converse notion is full-abstraction in the sense that $\mathbf{d}(\sigma_1(t), \sigma_2(t)) = \mathbf{A}([\![P]\!], \mathbf{d}(\sigma_1, \sigma_2))$ (no overapproximation). As demonstrated in Example 5.7, the approximation functionals would require for process variables $x \in \mathsf{Var}(t)$ besides the bisimulation distance between $\sigma_1(x)$ and $\sigma_2(x)$ also information about the reactive behavior and the branching. However, for our objective to study the distance of composed processes in relation to the distance of its components, the bisimulation distance is the right level of abstraction.

We conclude this section by introducing an ordering relation on $\mathscr{D}$ based on the ordering over $\mathscr{P}$ (Definition 5.17) that ensures monotonicity of both the approximation functional $\mathbf{A}$ and the functional $\mathbf{F}$ introduced in the next section to compute the denotation of arbitrary terms of a PGSOS PTSS.

Definition 5.24. We order the elements of $\mathscr{D}$ by $\sqsubseteq \subseteq \mathscr{D} \times \mathscr{D}$ defined as

$$P_1 \sqsubseteq P_2 \text{ iff for all } p_1 \in P_1 \text{ there is a } p_2 \in P_2 \text{ such that } p_1 \sqsubseteq p_2$$

for all $P_1, P_2 \in \mathscr{D}$.

It is clear that $\mathbf{A}(P, e) = 0$ if $P = 0 \in \mathscr{D}$ or $e = 0 \in \mathscr{E}$.

Proposition 5.25. *$(\mathscr{D}, \sqsubseteq)$ is a cpo with least element* 0.

Proof. Reflexivity follows trivially from the fact that relation $\sqsubseteq$ on $\mathscr{P}$ is reflexive. In order to show transitivity, assume $P_1, P_2, P_3 \in \mathscr{D}$ with $P_1 \sqsubseteq P_2$ and $P_2 \sqsubseteq P_3$, i.e. for each $p_1 \in P_1$ there is a $p_2 \in P_2$ with $p_1 \sqsubseteq p_2$, and for each $p_2 \in P_2$ there is a $p_3 \in P_3$ with $p_2 \sqsubseteq p_3$. By transitivity of $\sqsubseteq$ on $\mathscr{P}$ we have then that for each $p_1 \in P_1$ there is a $p_3 \in P_3$ with $p_1 \sqsubseteq p_3$.

Hence, $P_1 \sqsubseteq P_3$. To show antisymmetry assume $P_1 \sqsubseteq P_2$ and $P_2 \sqsubseteq P_1$ for $P_1, P_2 \in \mathscr{D}$. From $P_1 \sqsubseteq P_2$, given any $p \in P_1$ there is some $p' \in P_2$ with $p \sqsubseteq p'$. Since $P_2 = {\downarrow}P_2$ it follows $p \in P_2$. Hence $P_1 \subseteq P_2$. Analogously, $P_2 \subseteq P_1$. Hence $P_1 = P_2$. Finally, the supremum of an ω-chain over $\mathscr{D}$ is trivially obtained by the union on its elements. We conclude by observing that it is clear that $0 \in \mathscr{D}$ is the least element of $(\mathscr{D}, \sqsubseteq)$. □

Proposition 5.26. *Let $P, P' \in \mathscr{D}$ and $e, e' \in \mathscr{E}$. Then*

1. $\mathbf{A}(P, e) \leq \mathbf{A}(P', e)$ *if* $P \sqsubseteq P'$;
2. $\mathbf{A}(P, e) \leq \mathbf{A}(P, e')$ *if* $e \sqsubseteq e'$.

Proof. To prove monotonicity of **A** we exploit monotonicity of **P** (Proposition 5.19). Consider first case 1. By $P \sqsubseteq P'$ we have that for all $p \in P$ there is a $p' \in P'$ such that $p \sqsubseteq p'$. By monotonicity of **P** on the first argument (Lemma 5.19.1) we get that $\mathbf{P}(p, e) \leq \mathbf{P}(p', e)$. Hence, $\mathbf{A}(P, e) = \sup_{p \in P} \mathbf{P}(p, e) \leq \sup_{p' \in P'} \mathbf{P}(p', e) = \mathbf{A}(P', e)$.

Consider now case 2. By monotonicity of **P** on the second argument (Lemma 5.19.2) we have $\mathbf{P}(p, e) \leq \mathbf{P}(p, e')$ for all $p \in P$. Hence, $\mathbf{A}(P, e) = \sup_{p \in P} \mathbf{P}(p, e) \leq \sup_{p \in P} \mathbf{P}(p, e') = \mathbf{A}(P, e')$. □

We will see in the following section that the denotations developed for terms of P_{PA} are sufficient for terms of any PGSOS PTSS.

5.3 Distance between composed processes

Now we generalize the method developed in the previous section to compute the denotation of open terms specified by arbitrary PGSOS PTSS. In line with the former section this gives an upper bound on the bisimulation distance between the closed instances of a given term. In particular, the denotation for the term[2] $f(x_1, \ldots, x_{\mathsf{r}(f)})$ gives an upper bound on the distance between processes composed by the process combinator f. This allows us in the next section to formulate a simple condition to decide if a process combinator is uniformly continuous, and hence if we can reason compositionally over processes combined by that process combinator.

5.3.1 Operations on process denotations

We start by defining some operations on process denotations that allow us to compute the denotation of process terms by induction over the term structure. We define the operations first on $\mathscr{M}$ and then lift them to $\mathscr{P}$ and $\mathscr{D}$.

The composition of two processes t_1 and t_2 which both may proceed requires that their multiplicities are summed up (cf. parallel composition in the prior section). We define the *summation* of multiplicities $\oplus \colon \mathscr{M} \times \mathscr{M} \to \mathscr{M}$ by

$$(m_1 \oplus m_2)(x) = m_1(x) + m_2(x) \text{ for all } x \in \mathscr{V}$$

[2] In this chapter we will denote the rank of a function explicitly by $\mathsf{r}(f)$ (in former sections usually denoted by n) to not confuse with the step count n in up-to-n bisimulation metric.

for all $m_1, m_2 \in \mathscr{M}$.

In order to define by structural induction the multiplicity of a term $f(t_1, \ldots, t_{\mathrm{r}(f)})$, we need an operation that composes the multiplicity denoting the operator f, namely the multiplicity of term $f(x_1, \ldots, x_{\mathrm{r}(f)})$, with the multiplicities of terms $t_1, \ldots, t_{\mathrm{r}(f)}$. We define the *pointed multiplication* of multiplicities $\odot_y : \mathscr{M} \times \mathscr{M} \to \mathscr{M}$ with respect to variable $y \in \mathscr{V}$ by

$$(m_1 \odot_y m_2)(x) = m_1(y) \cdot m_2(x) \text{ for all } x \in \mathscr{V}$$

for all $m_1, m_2 \in \mathscr{M}$.

Then, the multiplicity of a deterministic state term $f(t_1, \ldots, t_{\mathrm{r}(f)})$ is obtained by combining the multiplicity of term $f(x_1, \ldots, x_{\mathrm{r}(f)})$ and the multiplicities of terms $t_1, \ldots, t_{\mathrm{r}(f)}$ by operations $\oplus$ and $\odot$ as follows:

$$[\![f(t_1, \ldots, t_{\mathrm{r}(f)})]\!]_{\mathscr{M}} = \bigoplus_{i=1}^{\mathrm{r}(f)} \big([\![f(x_1, \ldots, x_{\mathrm{r}(f)})]\!]_{\mathscr{M}} \odot_{x_i} [\![t_i]\!]_{\mathscr{M}} \big)$$

Example 5.27. Consider the open term $t = a.x \parallel y$. From Section 5.2 we get $[\![a.x]\!]_{\mathscr{M}} = (\lambda)_x$, $[\![y]\!]_{\mathscr{M}} = 1_y$ and $[\![x_1 \parallel x_2]\!]_{\mathscr{M}} = 1_{\{x_1, x_2\}}$. Then, we have $[\![t]\!]_{\mathscr{M}} = ([\![x_1 \parallel x_2]\!]_{\mathscr{M}} \odot_{x_1} [\![a.x]\!]_{\mathscr{M}}) \oplus ([\![x_1 \parallel x_2]\!]_{\mathscr{M}} \odot_{x_2} [\![y]\!]_{\mathscr{M}}) = (1_{\{x_1, x_2\}} \odot_{x_1} (\lambda)_x) \oplus (1_{\{x_1, x_2\}} \odot_{x_2} 1_y) = (\lambda)_x \oplus 1_y$.

The multiplicity of term t needs to weight (by the discount factor) those instances of variables in t that are spawned just when t has already evolved. The respective operation is the scalar multiplication $\cdot : \mathbb{R}_{>0} \times \mathscr{M} \to \mathscr{M}$ defined by

$$(\lambda \cdot m)(x) = \lambda \cdot m(x) \text{ for all } x \in \mathscr{V}$$

for all $\lambda \in \mathbb{R}$ and $m \in \mathscr{M}$.

Given an operator f defined by a PGSOS rules r, the three operations $\oplus$, $\odot$ and $\cdot$ on multiplicities allow us now to define the multiplicity of term $f(x_1, \ldots, x_{\mathrm{r}(f)})$ in terms of the multiplicity of the target of r. In detail, the multiplicity of $f(x_1, \ldots, x_{\mathrm{r}(f)})$ is defined in terms of the multiplicity of the rule r, which, in turn, is defined in terms of the multiplicity of its target $\mathrm{trgt}(r)$. Let μ be a derivative of the source variable x in rule r. In order to express the multiplicity $[\![\mathrm{trgt}(r)]\!]_{\mathscr{M}}(\mu)$ of μ in the target of r as a multiplicity of x in r, we use the property $([\![\mathrm{trgt}(r)]\!]_{\mathscr{M}} \odot_\mu 1_x)(x) = [\![\mathrm{trgt}(r)]\!]_{\mathscr{M}}(\mu)$. Then, the multiplicity of r is defined for any source variable x as the sum of the multiplicity of x, discounted by discount factor λ, and all the derivatives of x in $\mathrm{trgt}(r)$:

$$[\![r]\!]_{\mathscr{M}} = (\lambda \cdot [\![\mathrm{trgt}(r)]\!]_{\mathscr{M}}) \oplus \Bigg(\bigoplus_{\substack{a_{i,m} \\ x_i \xrightarrow{\quad} \mu_{i,m} \in \\ \mathrm{pprem}(r)}} [\![\mathrm{trgt}(r)]\!]_{\mathscr{M}} \odot_{\mu_{i,m}} 1_{x_i} \Bigg).$$

Then, the multiplicity of $f(x_1, \ldots, x_{\mathrm{r}(f)})$ is obtained from the multiplicity of r by considering only the multiplicities of the variables $x_1, \ldots, x_{\mathrm{r}(f)}$ by

$$[\![f(x_1, \ldots, x_{\mathrm{r}(f)})]\!]_{\mathscr{M}} = \bigoplus_{i=1}^{\mathrm{r}(f)} [\![r]\!]_{\mathscr{M}} \odot_{x_i} 1_{x_i}.$$

Example 5.28. Consider the open term $t = f(x)$ and the following rule r:

$$\frac{x \xrightarrow{a} \mu}{f(x) \xrightarrow{a} \theta}$$

with $\theta \in \mathbb{DT}(\Sigma)$ some distribution term. Consider now the closed substitutions $\sigma_1(x) = a.a.a.0$ and $\sigma_2(x) = a.([0.9]a.a.0 \oplus [0.1]0)$ with $\mathbf{d}(\sigma_1(x), \sigma_2(x)) = 0.1\lambda$. The process distance induced by σ_1 and σ_2 is then $\mathbf{d}(\sigma_1, \sigma_2)(x) = 0.1\lambda$ and $\mathbf{d}(\sigma_1, \sigma_2)(\mu) = 0$.

Consider $\theta = \mu \parallel \mu$. Similar to Example 5.4 we get $\mathbf{d}(\sigma_1(t), \sigma_2(t)) = \lambda(1-(1-\mathbf{d}(\sigma_1(x), \sigma_2(x))/\lambda)^2) = \lambda(1-(1-0.1)^2)$. The denotation of the target of r is $[\![\mathrm{trgt}(r)]\!]_{\mathscr{M}} = 2_\mu$. Hence, the denotation of r is $[\![r]\!]_{\mathscr{M}} = \lambda \cdot 2_\mu \oplus (2_\mu \odot_\mu 1_x) = (2\lambda)_\mu \oplus 2_x$. Thus, the denotation of term t is $[\![t]\!]_{\mathscr{M}} = ((2\lambda)_\mu \oplus 2_x) \odot_x 1_x = 2_x$. Therefore, $\mathbf{D}([\![t]\!]_{\mathscr{M}}, \mathbf{d}(\sigma_1, \sigma_2)) = \lambda(1-(1-0.1)^2) = \mathbf{d}(\sigma_1(t), \sigma_2(t))$.

Consider $\theta = \delta(x) \parallel \mu$. We get $\mathbf{d}(\sigma_1(t), \sigma_2(t)) = \lambda(1-(1-\mathbf{d}(\sigma_1(x), \sigma_2(x)))(1-\mathbf{d}(\sigma_1(x), \sigma_2(x))/\lambda)) = \lambda(1-(1-0.1\lambda)(1-0.1))$. The denotation of the target of r is $[\![\mathrm{trgt}(r)]\!]_{\mathscr{M}} = 1_{\{x,\mu\}}$. Hence, the denotation of r is $[\![r]\!]_{\mathscr{M}} = \lambda \cdot 1_{\{x,\mu\}} \oplus (1_{\{x,\mu\}} \odot_\mu 1_x) = (\lambda)_{\{x,\mu\}} + 1_x = (1+\lambda)_x \oplus \lambda_\mu$. Thus, the denotation of term t is $[\![t]\!]_{\mathscr{M}} = ((1+\lambda)_x \oplus \lambda_\mu) \odot_x 1_x = (1+\lambda)_x$. Therefore, $\mathbf{D}([\![t]\!]_{\mathscr{M}}, \mathbf{d}(\sigma_1, \sigma_2)) = \lambda(1-(1-0.1)^{1+\lambda}) \geq \mathbf{d}(\sigma_1(t), \sigma_2(t))$ (by Bernoulli's inequality, cf. Remark 5.15).

Consider $\theta = \delta(x) \parallel \delta(x)$. We get $\mathbf{d}(\sigma_1(t), \sigma_2(t)) = \lambda^2(1-(1-\mathbf{d}(\sigma_1(x), \sigma_2(x))/\lambda)(1-\mathbf{d}(\sigma_1(x), \sigma_2(x))/\lambda)) = \lambda(1-(1-\mathbf{d}(\sigma_1(x), \sigma_2(x)))(1-\mathbf{d}(\sigma_1(x), \sigma_2(x)))) - (1-\lambda) \cdot \mathbf{d}(\sigma_1(x), \sigma_2(x)) \leq \lambda(1-(1-\mathbf{d}(\sigma_1(x), \sigma_2(x)))(1-\mathbf{d}(\sigma_1(x), \sigma_2(x))))$. The denotation of the target of r is $[\![\mathrm{trgt}(r)]\!]_{\mathscr{M}} = 2_x$. Hence, the denotation of r is $[\![r]\!]_{\mathscr{M}} = \lambda \cdot 2_x \oplus (2_x \odot_\mu 1_x) = (2\lambda)_x$. Thus, the denotation of term t is $[\![t]\!]_{\mathscr{M}} = (2\lambda)_x \odot_x 1_x = (2\lambda)_x$. Therefore, $\mathbf{D}([\![t]\!]_{\mathscr{M}}, \mathbf{d}(\sigma_1, \sigma_2)) = \lambda(1-(1-0.1)^{2\lambda}) \geq \mathbf{d}(\sigma_1(t), \sigma_2(t))$.

Finally, we need an operation on multiplicities to define for multiple nondeterministic choices a multiplicity that covers the multiplicity of each choice. For this purpose we define the *supremum* of multiplicities $\Cup\colon \mathscr{M} \times \mathscr{M} \to \mathscr{M}$ by:

$$m_1 \Cup m_2 = \sup\{m_1, m_2\}$$

for all $m_1, m_2 \in \mathscr{M}$.

The operations $op \in \{\oplus, \odot_y, \Cup\}\colon \mathscr{M} \times \mathscr{M} \to \mathscr{M}$ lift to $op\colon \mathscr{P} \times \mathscr{P} \to \mathscr{P}$ and $op\colon \mathscr{D} \times \mathscr{D} \to \mathscr{D}$ by

$$\begin{aligned} (p_1 \mathbin{op} p_2)(m) &= \sum_{\substack{m_1, m_2 \in \mathscr{M} \\ m = m_1 \mathbin{op} m_2}} p_1(m_1) \cdot p_2(m_2) \\ p \in (P_1 \mathbin{op} P_2) &\text{ iff } \exists p_1 \in P_1. \exists p_2 \in P_2. p \sqsubseteq p_1 \mathbin{op} p_2 \end{aligned} \tag{5.2}$$

for all $p_1, p_2 \in \mathscr{P}$ and $P_1, P_2 \in \mathscr{D}$.

Analogously, $\cdot\colon \mathbb{R}_{>0} \times \mathscr{M} \to \mathscr{M}$ lifts to $\cdot\colon \mathbb{R}_{>0} \times \mathscr{P} \to \mathscr{P}$ and $\cdot\colon \mathbb{R}_{>0} \times \mathscr{D} \to \mathscr{D}$ by

$$\begin{aligned} (r \cdot p)(r \cdot m) &= p(m) \\ r \cdot p \in (r \cdot P) &\text{ iff } p \in P \end{aligned} \tag{5.3}$$

for all $p \in \mathscr{P}$ and $P \in \mathscr{D}$. We remark that the lifting in Equation 5.3 has a simpler formulation than the lifting in Equation 5.2 since the scalar multiplication with a non-zero finite real is a bijective function.

In order to define the multiplicity of a rule having as target a convex combination of distribution terms, we introduce the *convex combinations of probabilistic multiplicities* $+ : \mathbb{R}_{\geq 0} \times \mathscr{P} \times \mathscr{P} \to \mathscr{P}$ defined by

$$(q \cdot p_1 + (1-q) \cdot p_2)(m) = qp_1(m) + (1-q)p_2(m) \text{ for all } m \in \mathscr{M}$$

for all $q \in \mathbb{R}$ and $p_1, p_2 \in \mathscr{P}$. Convex combination operator is lifted to nondeterministic probabilistic multiplicities by

$$p \in (q \cdot P_1 + (1-q) \cdot P_2) \text{ iff } \exists p_1 \in P_1, p_2 \in P_2 . p \sqsubseteq q \cdot p_1 + (1-q) \cdot p_2$$

for all $P_1, P_2 \in \mathscr{D}$.

5.3.2 Properties of operations on process denotations

Operators $\oplus$ and $\mathbin{\textcircled{\vee}}$ over $\mathscr{M}$ are commutative, associative and have unit element 0. Operator $\odot_y$ is associative but not commutative, and it distributes over $\oplus$ and $\mathbin{\textcircled{\vee}}$.

Lemma 5.29. *Let $m_1, m_2, m_3 \in \mathscr{M}$ and $y, y_1, y_2 \in \mathscr{V}$. Then:*

$$\begin{aligned}
m_1 \oplus m_2 &= m_2 \oplus m_1 \\
m_1 \oplus (m_2 \oplus m_3) &= (m_1 \oplus m_2) \oplus m_3 \\
0 \oplus m &= m \\
0 \odot_y m &= 0 \\
m \odot_y 0 &= 0 \\
1_y \odot_y m &= m \\
(m_1 \odot_{y_1} m_2) \odot_{y_2} m_3 &= m_1 \odot_{y_1} (m_2 \odot_{y_2} m_3) \\
m_1 \odot_y (m_2 \oplus m_3) &= (m_1 \odot_y m_2) \oplus (m_1 \odot_y m_3) \\
(m_1 \oplus m_2) \odot_y m_3 &= (m_1 \odot_y m_3) \oplus (m_2 \odot_y m_3) \\
m_1 \mathbin{\textcircled{\vee}} m_2 &= m_2 \mathbin{\textcircled{\vee}} m_1 \\
m_1 \mathbin{\textcircled{\vee}} (m_2 \mathbin{\textcircled{\vee}} m_3) &= (m_1 \mathbin{\textcircled{\vee}} m_2) \mathbin{\textcircled{\vee}} m_3 \\
0 \mathbin{\textcircled{\vee}} m &= m \\
m_1 \mathbin{\textcircled{\vee}} (m_2 \oplus m_3) &\leq (m_1 \mathbin{\textcircled{\vee}} m_2) \oplus (m_1 \mathbin{\textcircled{\vee}} m_3) \\
(m_1 \mathbin{\textcircled{\vee}} m_2) \odot_y m_3 &= (m_1 \odot_y m_3) \mathbin{\textcircled{\vee}} (m_2 \odot_y m_3) \\
m_1 \odot_y (m_2 \mathbin{\textcircled{\vee}} m_3) &= (m_1 \odot_y m_2) \mathbin{\textcircled{\vee}} (m_1 \odot_y m_3)
\end{aligned}$$

Proof. Straightforward. □

Furthermore, the lifting to $\mathscr{P}$ and $\mathscr{D}$ in equation 5.2 preserves commutativity, associativity and idempotency of the operators. The nondeterministic probabilistic multiplicity $0 \in \mathscr{D}$ is the left unit element for $\oplus$ and $\mathbin{\textcircled{\vee}}$, $1_y \in \mathscr{D}$ is the unit element of $\odot_y$, and $1 \in \mathbb{R}_{>0}$ is the left unit element of $\cdot$.

The following lemma shows how operators $\oplus$, $\odot$ and $\mathbin{\textcircled{\vee}}$ over $\mathscr{M}$ distribute over the deterministic distance approximation functional **D** and how **D** applies to a multiplicity of the form $r \cdot m$.

Lemma 5.30. *Let $m_1, m_2 \in \mathscr{M}$ and $e \in \mathscr{E}$ with $e(x) < 1$ for all $x \in \mathscr{V}$. Then*

1. $\mathbf{D}(m_1 \oplus m_2, e) = \lambda \cdot \left(1 - \left(1 - \frac{\mathbf{D}(m_1,e)}{\lambda}\right)\left(1 - \frac{\mathbf{D}(m_2,e)}{\lambda}\right)\right)$.
2. $\mathbf{D}(m_1 \odot_y m_2, e) = \lambda \cdot \left(1 - \left(1 - \frac{\mathbf{D}(m_2,e)}{\lambda}\right)^{m_1(y)}\right)$.
3. $\mathbf{D}(m_1 \mathbin{\textcircled{\scriptsize\vee}} m_2, e) = \lambda \cdot \left(1 - \prod_{x \in \mathscr{V}} \left(1 - \frac{e(x)}{\lambda}\right)^{\sup(m_1(x), m_2(x))}\right)$.
4. $\mathbf{D}(r \cdot m_1, e) = \lambda \cdot \left(1 - \prod_{x \in \mathscr{V}} \left(1 - \frac{e(x)}{\lambda}\right)^{r m_1(x)}\right)$.

Proof. We have:

1.
$$\begin{aligned}
&\mathbf{D}(m_1 \oplus m_2, e)\\
=&\lambda \cdot \left(1 - \prod_{x \in \mathscr{V}} \left(1 - \frac{e(x)}{\lambda}\right)^{(m_1 \oplus m_2)(x)}\right)\\
=&\lambda \cdot \left(1 - \prod_{x \in \mathscr{V}} \left(1 - \frac{e(x)}{\lambda}\right)^{m_1(x)} \cdot \prod_{x \in \mathscr{V}} \left(1 - \frac{e(x)}{\lambda}\right)^{m_2(x)}\right)\\
=&\lambda \cdot \left(1 - \left(1 - \left(1 - \prod_{x \in \mathscr{V}} \left(1 - \frac{e(x)}{\lambda}\right)^{m_1(x)}\right)\right) \cdot \right.\\
&\qquad \left.\left(1 - \left(1 - \prod_{x \in \mathscr{V}} \left(1 - \frac{e(x)}{\lambda}\right)^{m_2(x)}\right)\right)\right)\\
=&\lambda \cdot \left(1 - \left(1 - \frac{\mathbf{D}(m_1,e)}{\lambda}\right)\left(1 - \frac{\mathbf{D}(m_2,e)}{\lambda}\right)\right).
\end{aligned}$$

2.
$$\begin{aligned}
&\mathbf{D}(m_1 \odot_y m_2, e)\\
=&\lambda \cdot \left(1 - \prod_{x \in \mathscr{V}} \left(1 - \frac{e(x)}{\lambda}\right)^{(m_1 \odot_y m_2)(x)}\right)\\
=&\lambda \cdot \left(1 - \prod_{x \in \mathscr{V}} \left(1 - \frac{e(x)}{\lambda}\right)^{m_1(y) \cdot m_2(x)}\right)\\
=&\lambda \cdot \left(1 - \prod_{x \in \mathscr{V}} \left(\left(1 - \frac{e(x)}{\lambda}\right)^{m_2(x)}\right)^{m_1(y)}\right)\\
=&\lambda \cdot \left(1 - \left(\prod_{x \in \mathscr{V}} \left(1 - \frac{e(x)}{\lambda}\right)^{m_2(x)}\right)^{m_1(y)}\right)\\
=&\lambda \cdot \left(1 - \left(1 - \left(1 - \prod_{x \in \mathscr{V}} \left(1 - \frac{e(x)}{\lambda}\right)^{m_2(x)}\right)\right)^{m_1(y)}\right)
\end{aligned}$$

$$=\lambda\cdot\left(1-\left(1-\frac{\mathbf{D}(m_2,e)}{\lambda}\right)^{m_1(y)}\right).$$

3.

$$\begin{aligned}&\mathbf{D}(m_1 \mathbin{\textcircled{\scriptsize\vee}} m_2, e)\\ =&\lambda\cdot\left(1-\prod_{x\in\mathscr{V}}\left(1-\frac{e(x)}{\lambda}\right)^{(m_1\mathbin{\textcircled{\scriptsize\vee}}m_2)(x)}\right)\\ =&\lambda\cdot\left(1-\prod_{x\in\mathscr{V}}\left(1-\frac{e(x)}{\lambda}\right)^{\sup(m_1(x),m_2(x))}\right).\end{aligned}$$

4.

$$\begin{aligned}&\mathbf{D}(r\cdot m_1, e)\\ =&\lambda\cdot\left(1-\prod_{x\in\mathscr{V}}\left(1-\frac{e(x)}{\lambda}\right)^{(r\cdot m_1)(x)}\right)\\ =&\lambda\cdot\left(1-\prod_{x\in\mathscr{V}}\left(1-\frac{e(x)}{\lambda}\right)^{r m_1(x)}\right).\end{aligned}$$

□

The following lemma shows how operators $\oplus$, $\odot$, $\mathbin{\textcircled{\scriptsize\vee}}$ over $\mathscr{P}$ distribute over the probabilistic distance approximation functional $\mathbf{P}$, and how $\mathbf{P}$ applies to probabilistic multiplicities of the form $r\cdot p$ and to convex combinations of probabilistic multiplicities.

Lemma 5.31. *Let $p_1, p_2 \in \mathscr{P}$, $e \in \mathscr{E}$ with $e(x) < 1$ for all $x \in \mathscr{V}$, and $w \in [0,1]$. Then*

1. $\mathbf{P}(p_1 \oplus p_2, e) = \sum_{m_1,m_2\in\mathscr{M}} p_1(m_1)\cdot p_2(m_2)\cdot\lambda\cdot\left(1-\left(1-\frac{\mathbf{D}(m_1,e)}{\lambda}\right)\left(1-\frac{\mathbf{D}(m_2,e)}{\lambda}\right)\right)$.
2. $\mathbf{P}(p_1 \odot_y p_2, e) = \sum_{m_1,m_2\in\mathscr{M}} p_1(m_1)\cdot p_2(m_2)\cdot\lambda\cdot\left(1-\left(1-\frac{\mathbf{D}(m_2,e)}{\lambda}\right)^{m_1(y)}\right)$.
3. $\mathbf{P}(p_1 \mathbin{\textcircled{\scriptsize\vee}} p_2, e) = \sum_{m_1,m_2\in\mathscr{M}} p_1(m_1)\cdot p_2(m_2)\cdot\lambda\cdot\left(1-\prod_{x\in\mathscr{V}}\left(1-\frac{e(x)}{\lambda}\right)^{\sup(m_1(x),m_2(x))}\right)$.
4. $\mathbf{P}(r\cdot p_1, e) = \sum_{m\in\mathscr{M}} p(\frac{1}{r}\cdot m)\,\mathbf{D}(m,e)$.
5. $\mathbf{P}(w\cdot p_1 + (1-w)\cdot p_2, e) = w\cdot\mathbf{P}(p_1,e) + (1-w)\cdot\mathbf{P}(p_2,e)$.

Proof. 1. By using Lemma 5.30.1 we get:

$$\begin{aligned}&\mathbf{P}(p_1\oplus p_2, e)\\ =&\sum_{m\in\mathscr{M}}(p_1\oplus p_2)(m)\cdot\mathbf{D}(m,e)\\ =&\sum_{m_1,m_2\in\mathscr{M}} p_1(m_1)\cdot p_2(m_2)\cdot\mathbf{D}(m_1\oplus m_2,e)\\ =&\sum_{m_1,m_2\in\mathscr{M}} p_1(m_1)\cdot p_2(m_2)\cdot\lambda\cdot\left(1-\left(1-\frac{\mathbf{D}(m_1,e)}{\lambda}\right)\left(1-\frac{\mathbf{D}(m_2,e)}{\lambda}\right)\right).\end{aligned}$$

2. By using Lemma 5.30.2 we get:

$$\begin{aligned}
&\mathbf{P}(p_1 \odot_y p_2, e)\\
&= \sum_{m\in\mathscr{M}} (p_1 \odot_y p_2)(m)\cdot \mathbf{D}(m,e)\\
&= \sum_{m_1,m_2\in\mathscr{M}} p_1(m_1)\cdot p_2(m_2)\cdot \mathbf{D}(m_1 \odot_y m_2, e)\\
&= \sum_{m_1,m_2} p_1(m_1)\cdot p_2(m_2)\cdot \lambda\left(1-\left(1-\frac{\mathbf{D}(m_2,e)}{\lambda}\right)^{m_1(y)}\right).
\end{aligned}$$

3. By using Lemma 5.30.3 we get:

$$\begin{aligned}
&\mathbf{P}(p_1 \mathbin{\mathbb{\triangledown}} p_2, e)\\
&= \sum_{m\in\mathscr{M}} (p_1 \mathbin{\mathbb{\triangledown}} p_2)(m)\cdot \mathbf{D}(m,e)\\
&= \sum_{m_1,m_2\in\mathscr{M}} p_1(m_1)\cdot p_2(m_2)\cdot \mathbf{D}(m_1 \mathbin{\mathbb{\triangledown}} m_2, e)\\
&= \sum_{m_1,m_2\in\mathscr{M}} p_1(m_1)\cdot p_2(m_2)\cdot \lambda\cdot\left(1-\prod_{x\in\mathscr{V}}\left(1-\frac{e(x)}{\lambda}\right)^{\sup(m_1(x),m_2(x))}\right).
\end{aligned}$$

4. We have

$$\begin{aligned}
&\mathbf{P}(r\cdot p_1, e)\\
&= \sum_{m\in\mathscr{M}} (r\cdot p_1)(m)\cdot \mathbf{D}(m,e)\\
&= \sum_{m\in\mathscr{M}} p_1(\frac{1}{r}\cdot m)\cdot \mathbf{D}(m,e).
\end{aligned}$$

5. We have

$$\begin{aligned}
&\mathbf{P}(w\cdot p_1 + (1-w)\cdot p_2, e)\\
&= \sum_{m\in\mathscr{M}} (w\cdot p_1 + (1-w)\cdot p_2)(m)\cdot \mathbf{D}(m,e)\\
&= w\sum_{m\in\mathscr{M}} p_1(m)\cdot \mathbf{D}(m,e) + (1-w)\sum_{m\in\mathscr{M}} p_2(m)\cdot \mathbf{D}(m,e)\\
&= w\cdot \mathbf{P}(p_1,e) + (1-w)\cdot \mathbf{P}(p_2,e).
\end{aligned}$$

□

We show now that all operations $\oplus, \odot_y, \mathbb{\triangledown}, \cdot, +$ defined above and the union are monotone and continuous on $\mathscr{D}$.

Proposition 5.32. *The operators $\oplus, \odot_y, \mathbb{\triangledown}, \cdot, +, \cup$ on $\mathscr{D}$ are order preserving.*

Proof. We have to prove that for arbitrary $P_1, P_2, P_1', P_2' \in \mathscr{D}$ with $P_1 \sqsubseteq P_1'$ and $P_2 \sqsubseteq P_2'$, and $q, q' \in [0,1]$ with $q \leq q'$,

1. $P_1 \oplus P_2 \sqsubseteq P_1' \oplus P_2'$.
2. $P_1 \odot_y P_2 \sqsubseteq P_1' \odot_y P_2'$.
3. $P_1 \mathbin{\mathbb{O}} P_2 \sqsubseteq P_1' \mathbin{\mathbb{O}} P_2'$.
4. $q \cdot P_1 \sqsubseteq q' \cdot P_1'$.
5. $qP_1 + (1-q)P_2 \sqsubseteq qP_1' + (1-q)P_2'$.
6. $P_1 \cup P_2 \sqsubseteq P_1' \cup P_2'$.

First we observe that monotonicity of operators $\oplus, \odot_y, \mathbb{O}, \cdot$ on $\mathscr{M}$ is immediate.

Now we show monotonicity of operators $\oplus, \odot_y, \mathbb{O}, \cdot, +$ on $\mathscr{P}$, from which monotonicity of the same operators on $\mathscr{D}$ will be derived. Let $q \leq q' \in \mathbb{R}$ and $p_1, p_2, p_1', p_2' \in \mathscr{P}$ with $p_1 \sqsubseteq p_1'$ and $p_2 \sqsubseteq p_2'$, i.e. there are two matchings $\omega_1 \in \Omega(p_1, p_1')$ and $\omega_2 \in \Omega(p_2, p_2')$ such that $m_1 \sqsubseteq m_1'$ for all $m_1, m_1' \in \mathscr{M}$ with $\omega_1(m_1, m_1') > 0$ and $m_2 \sqsubseteq m_2'$ for all $m_2, m_2' \in \mathscr{M}$ with $\omega_2(m_2, m_2') > 0$.

Consider first operator $\oplus$. We need to show $p_1 \oplus p_2 \sqsubseteq p_1' \oplus p_2'$. By definition we have $(p_1 \oplus p_2)(m) = \sum_{m = m_1 \oplus m_2} p_1(m_1) \cdot p_2(m_2)$ and $(p_1' \oplus p_2')(m) = \sum_{m = m_1 \oplus m_2} p_1'(m_1) \cdot p_2'(m_2)$. Let $\omega \in \Delta(\mathscr{M} \times \mathscr{M})$ be the distribution defined by

$$\omega(m, m') = \sum_{\substack{m_1, m_2, m_1', m_2' \in \mathscr{M} \\ m = m_1 \oplus m_2 \\ m' = m_1' \oplus m_2'}} \omega_1(m_1, m_1') \cdot \omega_2(m_2, m_2') \tag{5.4}$$

for all $m, m' \in \mathscr{M}$. We have to prove that ω is a matching $\omega \in \Omega(p_1 \oplus p_2, p_1' \oplus p_2')$ for $p_1 \oplus p_2$ and $p_1' \oplus p_2'$ such that $m \sqsubseteq m'$ for all $m, m' \in \mathscr{M}$ with $\omega(m, m') > 0$. Notice that for $m = m_1 \oplus m_2$ we have

$$\sum_{m' \in \mathscr{M}} \sum_{\substack{m_1', m_2' \in \mathscr{M} \\ m' = m_1' \oplus m_2'}} \omega_1(m_1, m_1') \cdot \omega_2(m_2, m_2') = \sum_{m_1' \in \mathscr{M}} \omega_1(m_1, m_1') \sum_{m_2' \in \mathscr{M}} \omega_2(m_2, m_2'). \tag{5.5}$$

Then:

$$\begin{aligned}
&\sum_{m' \in \mathscr{M}} \omega(m, m') \\
&= \sum_{m' \in \mathscr{M}} \sum_{\substack{m_1, m_2, m_1', m_2' \in \mathscr{M} \\ m = m_1 \oplus m_2 \\ m' = m_1' \oplus m_2'}} \omega_1(m_1, m_1') \cdot \omega_2(m_2, m_2') \\
&= \sum_{\substack{m_1, m_2 \in \mathscr{M} \\ m = m_1 \oplus m_2}} \sum_{m' \in \mathscr{M}} \sum_{\substack{m_1', m_2' \in \mathscr{M} \\ m' = m_1' \oplus m_2'}} \omega_1(m_1, m_1') \cdot \omega_2(m_2, m_2') \\
&= \sum_{\substack{m_1, m_2 \in \mathscr{M} \\ m = m_1 \oplus m_2}} \left(\sum_{m_1' \in \mathscr{M}} \omega_1(m_1, m_1') \sum_{m_2' \in \mathscr{M}} \omega_2(m_2, m_2') \right) \qquad \text{(by Equation. 5.5)}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{\substack{m_1,m_2\in\mathscr{M}\\ m=m_1\oplus m_2}} p_1(m_1)\cdot p_2(m_2)\\
&=(p_1\oplus p_2)(m)
\end{aligned}
$$

and, analogously, $\sum_{m\in\mathscr{M}} \omega(m,m') = (p_1'\oplus p_2')(m')$, thus implying that ω is a matching $\omega\in\Omega(p_1\oplus p_2, p_1'\oplus p_2')$ for $p_1\oplus p_2$ and $p_1'\oplus p_2'$. It remains to prove that $\omega(m,m')>0$ implies $m\sqsubseteq m'$. Assume $\omega(m,m')>0$. Then there exist $m_1,m_2,m_1',m_2'\in\mathscr{M}$ with $m=m_1\oplus m_2$, $m'=m_1'\oplus m_2'$ and $\omega_1(m_1,m_1'),\omega_2(m_2,m_2')>0$. From $\omega_1(m_1,m_1'),\omega_2(m_2,m_2')>0$ it follows $m_1\sqsubseteq m_1'$ and $m_2\sqsubseteq m_2'$, thus implying $m\sqsubseteq m'$ by the monotonicity of $\oplus$ over $\mathscr{M}$.

Consider now operator $\odot_y$. We need to show $p_1\odot_y p_2\sqsubseteq p_1'\odot_y p_2'$. By definition we have $(p_1\odot_y p_2)(m)=\sum_{m=m_1\odot_y m_2} p_1(m_1)\cdot p_2(m_2)$ and $(p_1'\odot_y p_2')(m)=\sum_{m=m_1\odot_y m_2} p_1'(m_1)\cdot p_2'(m_2)$. Let $\omega\in\Delta(\mathscr{M}\times\mathscr{M})$ be the distribution defined by

$$
\omega(m,m') = \sum_{\substack{m_1,m_2,m_1',m_2'\in\mathscr{M}\\ m=m_1\odot_y m_2\\ m'=m_1'\odot_y m_2'}} \omega_1(m_1,m_1')\cdot\omega_2(m_2,m_2') \tag{5.6}
$$

for all $m,m'\in\mathscr{M}$. We have to prove that ω is a matching $\omega\in\Omega(p_1\odot_y p_2, p_1'\odot_y p_2')$ for $p_1\odot_y p_2$ and $p_1'\odot_y p_2'$ such that $m\sqsubseteq m'$ for all $m,m'\in\mathscr{M}$ with $\omega(m,m')>0$. This follows just like in case $\oplus$.

Consider now operator $\mathbin{\mathbb{O}}$. We need to show $p_1\mathbin{\mathbb{O}} p_2\sqsubseteq p_1'\mathbin{\mathbb{O}} p_2'$. By definition we have $(p_1\mathbin{\mathbb{O}} p_2)(m)=\sum_{m=m_1\mathbin{\mathbb{O}} m_2} p_1(m_1)\cdot p_2(m_2)$ and $(p_1'\mathbin{\mathbb{O}} p_2')(m)=\sum_{m=m_1\mathbin{\mathbb{O}} m_2} p_1'(m_1)\cdot p_2'(m_2)$. Let $\omega\in\Delta(\mathscr{M}\times\mathscr{M})$ be the distribution defined by

$$
\omega(m,m') = \sum_{\substack{m_1,m_2,m_1',m_2'\in\mathscr{M}\\ m=m_1\mathbin{\mathbb{O}} m_2\\ m'=m_1'\mathbin{\mathbb{O}} m_2'}} \omega_1(m_1,m_1')\cdot\omega_2(m_2,m_2') \tag{5.7}
$$

for all $m,m'\in\mathscr{M}$. We have to prove that ω is a matching $\omega\in\Omega(p_1\mathbin{\mathbb{O}} p_2, p_1'\mathbin{\mathbb{O}} p_2')$ for $p_1\mathbin{\mathbb{O}} p_2$ and $p_1'\mathbin{\mathbb{O}} p_2'$ such that $m\sqsubseteq m'$ for all $m,m'\in\mathscr{M}$ with $\omega(m,m')>0$. This follows just like in case $\oplus$.

Consider now operator $\cdot$. We need to show $q\cdot p_1\sqsubseteq q'\cdot p_1'$. By definition we have $(q\cdot p_1)(q\cdot m)=p_1(m)$ and $(q'\cdot p_1')(q'\cdot m')=p_1'(m')$. Let $\omega\in\Delta(\mathscr{M}\times\mathscr{M})$ be the distribution defined by

$$
\omega(q\cdot m, q'\cdot m') = \omega_1(m,m') \tag{5.8}
$$

for all $m,m'\in\mathscr{M}$. We have to prove that ω is a matching $\omega\in\Omega(q\cdot p_1, q'\cdot p_1')$ for $q\cdot p_1$ and $q'\cdot p_1'$ such that $m\sqsubseteq m'$ for all $m,m'\in\mathscr{M}$ with $\omega(m,m')>0$. We have

$$
\begin{aligned}
&\sum_{m'\in\mathscr{M}} \omega(q\cdot m, q'\cdot m')\\
&= \sum_{m'\in\mathscr{M}} \omega_1(m,m')\\
&=p_1(m)\\
&=(q\cdot p_1)(q\cdot m)
\end{aligned}
$$

and, analogously, $\sum_{m\in\mathscr{M}} \omega(q\cdot m, q'\cdot m') = (q'\cdot p'_1)(q'\cdot m')$, thus confirming that ω is a matching $\omega \in \Omega(q\cdot p_1, q'\cdot p'_1)$ for $q\cdot p_1$ and $q'\cdot p'_1$. It remains to prove that $\omega(q\cdot m, q'\cdot m') > 0$ implies $q\cdot m \sqsubseteq q'\cdot m'$. If $\omega(q\cdot m, q'\cdot m') > 0$ then $\omega_1(m,m') > 0$, which implies $m \sqsubseteq m'$. Then, $q \leq q'$ and $m \sqsubseteq m'$ gives $q\cdot m \sqsubseteq q'\cdot m'$ by monotonicity of $\cdot$ on $\mathscr{M}$.

We conclude with operator $+$. We need to show $qp_1 + (1-q)p_2 \sqsubseteq qp'_1 + (1-q)p'_2$. By definition we have $(qp_1+(1-q)p_2)(m) = qp_1(m)+(1-q)p_2(m)$ and $(qp'_1+(1-q)p'_2)(m) = qp'_1(m) + (1-q)p'_2(m)$. Let $\omega \in \Delta(\mathscr{M}\times\mathscr{M})$ be the distribution defined by

$$\omega(m,m') = q\omega_1(m,m') + (1-q)\omega_2(m,m') \tag{5.9}$$

for all $m, m' \in \mathscr{M}$. We have to prove that ω is a matching $\omega \in \Omega(qp_1 + (1-q)p_2, qp'_1 + (1-q)p'_2)$ for $qp_1 + (1-q)p_2$ and $qp'_1 + (1-q)p'_2$ such that $m \sqsubseteq m'$ for all $m, m' \in \mathscr{M}$ with $\omega(m,m') > 0$. We have

$$\begin{aligned}
&\sum_{m'\in\mathscr{M}} \omega(m,m') \\
=&\sum_{m'\in\mathscr{M}} (q\omega_1(m,m') + (1-q)\omega_2(m,m')) \\
=&qp_1(m) + (1-q)p_2(m) \\
=&(qp_1 + (1-q)p_2)(m)
\end{aligned}$$

and, analogously, $\sum_{m\in\mathscr{M}} \omega(m,m') = (qp'_1 + (1-q)p'_2)(m')$, thus confirming that ω is a matching $\omega \in \Omega(qp_1 + (1-q)p_2, qp'_1 + (1-q)p'_2)$ for $qp_1 + (1-q)p_2$ and $qp'_1 + (1-q)p'_2$. Then, $\omega(m,m') > 0$ implies $\omega_1(m,m') > 0$ or $\omega_2(m,m') > 0$, thus implying $m \sqsubseteq m'$.

We proceed to show that operators $\oplus, \odot_y, \mathbb{O}, \cdot, +$ on $\mathscr{D}$ are monotone. We start with operators in $\{\oplus, \odot_y, \mathbb{O}, +\}$, namely we prove properties 1, 2, 3, 5. Let $\bowtie \in \{\oplus, \odot_y, \mathbb{O}, +\}$. We need to show that for each $p \in P_1 \bowtie P_2$ there is a $p' \in P'_1 \bowtie P'_2$ with $p \sqsubseteq p'$. By definition, $p \in P_1 \bowtie P_2$ implies that $p \sqsubseteq p_1 \bowtie p_2$ for some $p_1 \in P_1$ and $p_2 \in P_2$. By assumption $P_1 \sqsubseteq P'_1$ and $P_2 \sqsubseteq P'_2$ there exist $p'_1 \in P'_1$ and $p'_2 \in P'_2$ with $p_1 \sqsubseteq p'_1$ and $p_2 \sqsubseteq p'_2$. By monotonicity of $\bowtie$ on $\mathscr{P}$ (see above) it follows that $p_1 \bowtie p_2 \sqsubseteq p'_1 \bowtie p'_2$. From $p'_1 \in P'_1$ and $p'_2 \in P'_2$ it follows $p'_1 \bowtie p'_2 \in P'_1 \bowtie P'_2$. Hence, $p' = p'_1 \bowtie p'_2$ is the required probabilistic multiplicity in $P'_1 \bowtie P'_2$ such that $p \sqsubseteq p'$.

Now we consider operator $\cdot$, namely we prove property 4. We need to show that for each $p \in q\cdot P_1$ there is a $p' \in q'\cdot P'_1$ with $p \sqsubseteq p'$. By definition, $p \in q\cdot P_1$ iff $p \sqsubseteq q\cdot p_1$ for some $p_1 \in P_1$. By assumption $P_1 \sqsubseteq P'_1$ there exists $p'_1 \in P'_1$ with $p_1 \sqsubseteq p'_1$. By monotonicity of $\cdot$ on $\mathscr{P}$ (see above) we have $q\cdot p_1 \sqsubseteq q'\cdot p'_1$. From $p'_1 \in P'_1$ it follows $q'\cdot p'_1 \in q'\cdot P'_1$. Hence $p' = q'\cdot p'_1$ is the required probabilistic multiplicity in $q'\cdot P'_1$ such that $p \sqsubseteq p'$.

Finally, monotonicity of union on $\mathscr{D}$, namely property 6, follows immediately from the definition of order $\sqsubseteq$ on $\mathscr{D}$. □

Proposition 5.33. *The operators $\oplus, \odot_y, \mathbb{O}, \cdot, +, \cup$ on $\mathscr{D}$ are upwardly ω-continuous.*

Proof. We start with showing upward ω-continuity of $\oplus, \odot_y, \mathbb{O}, \cdot, +$ on $\mathscr{P}$, from which upward ω-continuity of the same operators on $\mathscr{D}$ will be derived. Let $(p_k)_{k\in\mathbb{N}}$ with $p_1 \sqsubseteq p_2 \sqsubseteq \ldots \in \mathscr{P}$ and $(p'_k)_{k\in\mathbb{N}}$ with $p'_1 \sqsubseteq p'_2 \sqsubseteq \ldots \in \mathscr{P}$, i.e. there are matchings $\omega_{(k,k+1)} \in \Omega(p_k, p_{k+1})$ and $\omega'_{(k,k+1)} \in \Omega(p'_k, p'_{k+1})$ such that $m_k \sqsubseteq m_{k+1}$ for all $m_k, m_{k+1} \in \mathscr{M}$ with

$\omega_{(k,k+1)}(m_k, m_{k+1}) > 0$ and $m'_k \sqsubseteq m'_{k+1}$ for all $m'_k, m'_{k+1} \in \mathscr{M}$ with $\omega'_{(k,k+1)}(m'_k, m'_{k+1}) > 0$. Let $p \in \mathscr{P}$. Let us recall that in the proof of Proposition 5.18 we get (Equation 5.1)

$$\sup(p_k)_{k\in\mathbb{N}}(m) = \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M} \\ \forall k'\le h.\exists m_h\in\mathscr{M}. \\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0 \\ m=\sup\{m_h \mid h\ge k'\}}} p_{k'}(m_{k'}) \tag{5.10}$$

for all $m \in \mathscr{M}$.

Consider first operator $\oplus$. We prove the upward ω-continuity in the first argument, the upward ω-continuity in the second argument is analogous. Hence we have to prove the existence of $\sup(p_k \oplus p)_{k\in\mathbb{N}}$. First we note that a matching $\hat{\omega}_{(k,k+1)} \in \Omega(p_k \oplus p, p_{k+1} \oplus p)$ witnessing $p_k \oplus p \sqsubseteq p_{k+1} \oplus p$ can be obtained by following the proof of monotonicity of operator $\oplus$ on $\mathscr{P}$ (Proposition 5.32) and by using the matching $\omega \in \Omega(p, p)$ defined by $\omega(m, m) = p(m)$ by instantiating Equation 5.4 as

$$\hat{\omega}_{(k,k+1)}(m'_k, m'_{k+1}) = \sum_{m\in\mathscr{M}} p(m) \sum_{\substack{m_k, m_{k+1}\in\mathscr{M} \\ m'_k = m_k \oplus m \\ m'_{k+1} = m_{k+1}\oplus m}} \omega_{(k,k+1)}(m_k, m_{k+1})$$

for all $m'_k, m'_{k+1} \in \mathscr{M}$. Now we have

$$\begin{aligned}
&(\sup(p_k)_{k\in\mathbb{N}} \oplus p)(m) \\
&= \sum_{\substack{m',m''\in\mathscr{M} \\ m=m'\oplus m''}} \sup(p_k)_{k\in\mathbb{N}}(m') \cdot p(m'') \\
&= \sum_{\substack{m',m''\in\mathscr{M} \\ m=m'\oplus m''}} \left(\lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M} \\ \forall k'\le h.\exists m_h\in\mathscr{M}. \\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0 \\ m'=\sup\{m_h \mid h\ge k'\}}} p_{k'}(m_{k'}) \right) \cdot p(m'') \qquad \text{(by Equation 5.10)} \\
&= \lim_{k'\to\infty} \sum_{\substack{m',m''\in\mathscr{M} \\ m=m'\oplus m''}} \sum_{\substack{m_{k'}\in\mathscr{M} \\ \forall k'\le h.\exists m_h\in\mathscr{M}. \\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0 \\ m'=\sup\{m_h \mid h\ge k'\}}} p_{k'}(m_{k'}) \cdot p(m'') \\
&= \lim_{k'\to\infty} \sum_{\substack{m_{k'},m''\in\mathscr{M} \\ \forall k'\le h.\exists m_h\in\mathscr{M}. \\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0 \\ m=\sup\{m_h\oplus m'' \mid h\ge k'\}\cup\{m''\}}} p_{k'}(m_{k'}) \cdot p(m'') \\
&= \lim_{k'\to\infty} \sum_{\substack{m_{k'},m''\in\mathscr{M} \\ \forall k'\le h.\exists \hat{m}_h\in\mathscr{M}. \\ \hat{m}_{k'} = m_{k'}\oplus m'' \\ \forall h\ge k'.\left(\exists m_h,m_{h+1}\in\mathscr{M}.\hat{m}_h = m_h\oplus m''\wedge\hat{m}_{h+1}=m_{h+1}\oplus m''\wedge\omega_{(h,h+1)}(m_h,m_{h+1})>0\right) \\ m=\sup\{\hat{m}_h \mid h\ge k'\}}} p_{k'}(m_{k'}) \cdot p(m'')
\end{aligned}$$

$$= \lim_{k' \to \infty} \sum_{\substack{m_{k'}, m'' \in \mathscr{M} \\ \forall k' \leq h. \exists \hat{m}_h \in \mathscr{M}. \\ \hat{m}_{k'} = m_{k'} \oplus m'' \\ \forall h \geq k'. \hat{\omega}_{(h,h+1)}(\hat{m}_h, \hat{m}_{h+1}) > 0 \\ m = \sup\{\hat{m}_h \mid h \geq k'\}}} p_{k'}(m_{k'}) \cdot p(m'')$$

$$= \lim_{k' \to \infty} \sum_{\substack{\hat{m}_{k'} \in \mathscr{M} \\ \forall k' \leq h. \exists \hat{m}_h \in \mathscr{M}. \\ \forall h \geq k'. \hat{\omega}_{(h,h+1)}(\hat{m}_h, \hat{m}_{h+1}) > 0 \\ m = \sup\{\hat{m}_h \mid h \geq k'\}}} \sum_{\substack{m_{k'}, m'' \in \mathscr{M} \\ \hat{m}_{k'} = m_{k'} \oplus m''}} p_{k'}(m_{k'}) \cdot p(m'')$$

$$= \lim_{k' \to \infty} \sum_{\substack{\hat{m}_{k'} \in \mathscr{M} \\ \forall k' \leq h. \exists \hat{m}_h \in \mathscr{M}. \\ \forall h \geq k'. \hat{\omega}_{(h,h+1)}(\hat{m}_h, \hat{m}_{h+1}) > 0 \\ m = \sup\{\hat{m}_h \mid h \geq k'\}}} (p_{k'} \oplus p)(\hat{m}_{k'})$$

$$= \sup(p_k \oplus p)_{k \in \mathbb{N}}(m) \qquad \text{(by Equation 5.10)}$$

thus confirming that $\sup(p_k \oplus p)_{k \in \mathbb{N}}$ exists.

Consider now operator $\odot$. We prove the upward ω-continuity in the first argument, the upward ω-continuity in the second argument is analogous. Hence we have to prove the existence of $\sup(p_k \odot_y p)_{k \in \mathbb{N}}$. First we note that a matching $\hat{\omega}_{(k,k+1)} \in \Omega(p_k \odot_y p, p_{k+1} \odot_y p)$ witnessing $p_k \odot_y p \sqsubseteq p_{k+1} \odot_y p$ can be obtained by following the proof of monotonicity of operator $\odot$ on $\mathscr{P}$ (Proposition 5.32) and by using the matching $\omega \in \Omega(p, p)$ defined by $\omega(m, m) = p(m)$ by instantiating Equation 5.6 as

$$\hat{\omega}_{(k,k+1)}(m'_k, m'_{k+1}) = \sum_{m \in \mathscr{M}} p(m) \sum_{\substack{m_k, m_{k+1} \in \mathscr{M} \\ m'_k = m_k \odot_y m \\ m'_{k+1} = m_{k+1} \odot_y m}} \omega_{(k,k+1)}(m_k, m_{k+1})$$

for all $m'_k, m'_{k+1} \in \mathscr{M}$. The remaining reasoning is just like in case of operator $\oplus$.

Consider now operator $\mathbin{\mathbb{O}}$. We prove the upward ω-continuity in the first argument, the upward ω-continuity in the second argument is analogous. Hence we have to prove the existence of $\sup(p_k \mathbin{\mathbb{O}}_y p)_{k \in \mathbb{N}}$. First we note that a matching $\hat{\omega}_{(k,k+1)} \in \Omega(p_k \mathbin{\mathbb{O}} p, p_{k+1} \mathbin{\mathbb{O}} p)$ witnessing $p_k \mathbin{\mathbb{O}} p \sqsubseteq p_{k+1} \mathbin{\mathbb{O}} p$ can be obtained by following the proof of monotonicity of operator $\mathbin{\mathbb{O}}$ on $\mathscr{P}$ (Proposition 5.32) and by using the matching $\omega \in \Omega(p, p)$ defined by $\omega(m, m) = p(m)$ by instantiating Equation 5.7 as

$$\hat{\omega}_{(k,k+1)}(m'_k, m'_{k+1}) = \sum_{m \in \mathscr{M}} p(m) \sum_{\substack{m_k, m_{k+1} \in \mathscr{M} \\ m'_k = \sup(m_k, m) \\ m'_{k+1} = \sup(m_{k+1}, m)}} \omega_{(k,k+1)}(m_k, m_{k+1})$$

for all $m'_k, m'_{k+1} \in \mathscr{M}$. The remaining reasoning is just like in case of operator $\oplus$.

Consider now operator $\cdot$. We have to prove the existence of $\sup(q \cdot p_k)_{k \in \mathbb{N}}$. First we note that a matching $\hat{\omega}_{(k,k+1)} \in \Omega(q \cdot p_k, q \cdot p_{k+1})$ witnessing $q \cdot p_k \sqsubseteq q \cdot p_{k+1}$ can be obtained by following the proof of monotonicity of operator $\cdot$ on $\mathscr{P}$ (Proposition 5.32) by instantiating Equation 5.8 as

$$\hat{\omega}_{(k,k+1)}(m_k, m_{k+1}) = \omega_{(k,k+1)}(1/q \cdot m_k, 1/q \cdot m_{k+1})$$

for all $m_k, m_{k+1} \in \mathscr{M}$. We have

$$
\begin{aligned}
&(q \cdot \sup(p_k)_{k\in\mathbb{N}})(m)\\
&= \sup(p_k)_{k\in\mathbb{N}}(1/q \cdot m) \quad \text{(by Equation 5.10)}\\
&= \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\le h.\exists m_h\in\mathscr{M}.\\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0\\ 1/q\cdot m=\sup\{m_h \mid h\ge k'\}}} p_{k'}(m_{k'})\\
&= \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\le h.\exists m_h\in\mathscr{M}.\\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0\\ m=\sup\{q\cdot m_h \mid h\ge k'\}}} p_{k'}(m_{k'})\\
&= \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\le h.\exists m_h\in\mathscr{M}.\\ \forall h\ge k'.\omega_{(h,h+1)}(1/q\cdot m_h,1/q\cdot m_{h+1})>0\\ m=\sup\{m_h \mid h\ge k'\}}} p_{k'}(1/q \cdot m_{k'})\\
&= \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\le h.\exists m_h\in\mathscr{M}.\\ \forall h\ge k'.\hat{\omega}_{(h,h+1)}(m_h,m_{h+1})>0\\ m=\sup\{m_h \mid h\ge k'\}}} p_{k'}(1/q \cdot m_{k'})\\
&= \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\le h.\exists m_h\in\mathscr{M}.\\ \forall h\ge k'.\hat{\omega}_{(h,h+1)}(m_h,m_{h+1})>0\\ m=\sup\{m_h \mid h\ge k'\}}} (q \cdot p_{k'})(m_{k'})\\
&= \sup(q \cdot p_k)_{k\in\mathbb{N}}(m) \quad \text{(by Equation 5.10)}
\end{aligned}
$$

thus confirming, by the arbitrarity of m, that $\sup(q \cdot p_k)_{k\in\mathbb{N}}$ exists.

Consider now operator $+$. We have to prove the existence of $\sup(qp_k + (1-q)p'_k)$. First we note that a matching $\hat{\omega}_{(k,k+1)} \in \Omega(qp_k + (1-q)p'_k, qp_{k+1} + (1-q)p'_{k+1})$ witnessing $qp_k + (1-q)p'_k \sqsubseteq qp_{k+1} + (1-q)p'_{k+1}$ can be obtained by following the proof of monotonicity of operator $\cdot$ on $\mathscr{P}$ (Proposition 5.32) by instantiating Equation 5.9 as

$$\hat{\omega}_{(k,k+1)}(m_k, m_{k+1}) = q\,\omega_{(k,k+1)}(m_k, m_{k+1}) + (1-q)\omega'_{(k,k+1)}(m_k, m_{k+1})$$

for all $m_m, m_{k+1} \in \mathscr{M}$. We have

$$
\begin{aligned}
&(q \sup(p_k)_{k\in\mathbb{N}} + (1-q)\sup(p'_k)_{k\in\mathbb{N}})(m)\\
&= q \sup(p_k)_{k\in\mathbb{N}}(m) + (1-q)\sup(p'_k)_{k\in\mathbb{N}}(m)\\
&= q \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\le h.\exists m_h\in\mathscr{M}.\\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0\\ m=\sup\{m_h \mid h\ge k'\}}} p_{k'}(m_{k'}) + (1-q) \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M}\\ \forall k'\le h.\exists m_h\in\mathscr{M}.\\ \forall h\ge k'.\omega'_{(h,h+1)}(m_h,m_{h+1})>0\\ m=\sup\{m_h \mid h\ge k'\}}} p'_{k'}(m_{k'}) \quad \text{(Eq. 5.10)}
\end{aligned}
$$

$$= \lim_{k'\to\infty} \left(q \sum_{\substack{m_{k'}\in\mathscr{M} \\ \forall k'\le h.\exists m_h\in\mathscr{M}. \\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0 \\ m=\sup\{m_h \mid h\ge k'\}}} p_{k'}(m_{k'}) + (1-q) \sum_{\substack{m_{k'}\in\mathscr{M} \\ \forall k'\le h.\exists m_h\in\mathscr{M}. \\ \forall h\ge k'.\omega'_{(h,h+1)}(m_h,m_{h+1})>0 \\ m=\sup\{m_h \mid h\ge k'\}}} p'_{k'}(m_{k'}) \right)$$

$$\le \lim_{k'\to\infty} \left(q \sum_{\substack{m_{k'}\in\mathscr{M} \\ \forall k'\le h.\exists m_h\in\mathscr{M}. \\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0\vee\omega'_{(h,h+1)}(m_h,m_{h+1})>0 \\ m=\sup\{m_h \mid h\ge k'\}}} p_{k'}(m_{k'}) + \right.$$

$$\left. (1-q) \sum_{\substack{m_{k'}\in\mathscr{M} \\ \forall k'\le h.\exists m_h\in\mathscr{M}. \\ \forall h\ge k'.\omega_{(h,h+1)}(m_h,m_{h+1})>0\vee\omega'_{(h,h+1)}(m_h,m_{h+1})>0 \\ m=\sup\{m_h \mid h\ge k'\}}} p'_{k'}(m_{k'}) \right)$$

$$= \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M} \\ \forall k'\le h.\exists m_h\in\mathscr{M}. \\ \forall h\ge k'.q\omega_{(h,h+1)}(m_h,m_{h+1})+(1-q)\omega'_{(h,h+1)}(m_h,m_{h+1})>0 \\ m=\sup\{m_h \mid h\ge k'\}}} (qp_{k'}(m_{k'}) + (1-q)p'_{k'}(m_{k'}))$$

$$= \lim_{k'\to\infty} \sum_{\substack{m_{k'}\in\mathscr{M} \\ \forall k'\le h.\exists m_h\in\mathscr{M}. \\ \forall h\ge k'.\hat{\omega}_{(h,h+1)}(m_h,m_{h+1})>0 \\ m=\sup\{m_h \mid h\ge k'\}}} (qp_{k'} + (1-q)p'_{k'})(m_{k'})$$

$$= \sup(qp_k + (1-q)p'_k)(m) \quad \text{(by Equation 5.10)}$$

thus confirming by the arbitrarity of m that $\sup(qp_k + (1-q)p'_k)$ exists.

We proceed to show that operators $\oplus, \odot_y, \mathbb{\odot}, \cdot, +$ on $\mathscr{D}$ are upwardly ω-continuous. We start by showing that $\oplus, \odot_y, \mathbb{\odot}$ are upwardly ω-continuous in their first argument. Let $\bowtie \in \{\oplus, \odot_y, \mathbb{\odot}\}$. We have to prove $\sup(P_k \bowtie P)_{k\in\mathbb{N}} = \sup(P_k)_{k\in\mathbb{N}} \bowtie P$. Relation $\sup(P_k \bowtie P)_{k\in\mathbb{N}} \sqsubseteq \sup(P_k)_{k\in\mathbb{N}} \bowtie P$ is immediate. We need to show $\sup(P_k)_{k\in\mathbb{N}} \bowtie P \sqsubseteq \sup(P_k \bowtie P)_{k\in\mathbb{N}}$, namely for each $p \in \sup(P_k)_{k\in\mathbb{N}} \bowtie P$ there exists some $p' \in \sup(P_k \bowtie P)_{k\in\mathbb{N}}$ with $p \sqsubseteq p'$. By definition, $p \in \sup(P_k)_{k\in\mathbb{N}} \bowtie P$ iff $p \sqsubseteq \hat{p} \bowtie \overline{p}$ for some $\hat{p} \in \sup(P_k)_{k\in\mathbb{N}}$ and $\overline{p} \in P$. By $\sup(P_k)_{k\in\mathbb{N}} = \bigcup_{k\in\mathbb{N}} P_k$ we infer that $\hat{p} \in P_k$ for some $k \in \mathbb{N}$. Hence $\hat{p} \bowtie \overline{p} \in P_k \bowtie P$ and, from $\sup_{k\in\mathbb{N}}(P_k \bowtie P) = \bigcup_{k\in\mathbb{N}}(P_k \bowtie P)$ we derive $\hat{p} \bowtie \overline{p} \in \sup_{k\in\mathbb{N}}(P_k \bowtie P)$. Hence $\hat{p} \bowtie \overline{p}$ is the p' we were looking for. The proof that $\oplus, \odot_y, \mathbb{\odot}$ are ω-upwardly continuous in their second argument is analogous.

We prove now that $\cdot$ is upwardly ω-continuous. We have to prove $\sup(q \cdot P_k)_{k\in\mathbb{N}} = q \cdot \sup(P_k)_{k\in\mathbb{N}}$. Relation $\sup(q \cdot P_k)_{k\in\mathbb{N}} \sqsubseteq q \cdot \sup(P_k)_{k\in\mathbb{N}}$ is immediate. We need to show $q \cdot \sup(P_k)_{k\in\mathbb{N}} \sqsubseteq \sup(q \cdot P_k)_{k\in\mathbb{N}}$, namely for each $p \in q \cdot \sup(P_k)_{k\in\mathbb{N}}$ there exists some $p' \in \sup(q \cdot P_k)_{k\in\mathbb{N}}$ with $p \sqsubseteq p'$. By definition, $p \in q \cdot \sup(P_k)_{k\in\mathbb{N}}$ iff $p \sqsubseteq q \cdot \hat{p}$ for some $\hat{p} \in \sup(P_k)_{k\in\mathbb{N}}$. By $\sup(P_k)_{k\in\mathbb{N}} = \bigcup_{k\in\mathbb{N}} P_k$ we infer that $\hat{p} \in P_k$ for some $k \in \mathbb{N}$. Hence

$q \cdot \hat{p} \in q \cdot P_k$ and, from $\sup_{k\in\mathbb{N}}(q \cdot P_k) = \bigcup_{k\in\mathbb{N}}(q \cdot P_k)$ we derive $q \cdot \hat{p} \in \sup_{k\in\mathbb{N}}(q \cdot P_k)$. Hence $q \cdot \hat{p}$ is the p' we were looking for.

We prove now that $+$ is upwardly ω-continuous. We need to show that $\sup(qP_k + (1-p)P'_k)_{k\in\mathbb{N}} = q\sup(P_k)_{k\in\mathbb{N}} + (1-q)\sup(P'_k)_{k\in\mathbb{N}}$. Relation $\sup(qP_k + (1-q)P'_k)_{k\in\mathbb{N}} \sqsubseteq q\sup(P_k)_{k\in\mathbb{N}} + (1-q)\sup(P'_k)_{k\in\mathbb{N}}$ is immediate. We need to show $q \cdot \sup(P_k)_{k\in\mathbb{N}} + (1-q)\sup(P'_k)_{k\in\mathbb{N}} \sqsubseteq \sup(qP_k + (1-q)P'_k)_{k\in\mathbb{N}}$, i.e. for each $p \in q\sup(P_k)_{k\in\mathbb{N}} + (1-q)\sup(P'_k)_{k\in\mathbb{N}}$ there is some $p' \in \sup(q \cdot P_k + (1-q)P'_k)_{k\in\mathbb{N}}$ with $p \sqsubseteq p'$. By definition, $p \in q \cdot \sup(P_k)_{k\in\mathbb{N}} + (1-q)\sup(P'_k)_{k\in\mathbb{N}}$ iff $p \sqsubseteq q \cdot \hat{p} + (1-q)\hat{p'}$ for some $\hat{p} \in \sup(P_k)_{k\in\mathbb{N}}$ and $\hat{p'} \in \sup(P'_k)_{k\in\mathbb{N}}$. By $\sup(P_k)_{k\in\mathbb{N}} = \bigcup_{k\in\mathbb{N}} P_k$ and $\sup(P'_k)_{k\in\mathbb{N}} = \bigcup_{k\in\mathbb{N}} P'_k$ we infer that $\hat{p} \in P_k$ and $\hat{p'} \in P'_{k'}$ for some $k, k' \in \mathbb{N}$. Therefore for $k, k' \leq k''$ there exist $p_1 \in P_{k''}$ and $p'_1 \in P'_{k''}$ with $\hat{p} \sqsubseteq p_1$ and $\hat{p'} \sqsubseteq p'_1$. Hence $qp_1 + (1-q)p_2 \in qP_{k''} + (1-q)P'_{k''}$. From $\sup_{k\in\mathbb{N}}(qP_k + (1-q)P'_k) = \bigcup_{k\in\mathbb{N}}(qP_k + (1-q)P'_k)$ we derive $qp_1 + (1-q)p_2 \in \sup(q \cdot P_k + (1-q)P'_k)_{k\in\mathbb{N}}$. Hence $qp_1 + (1-q)p_2$ is the p' we were looking for.

Upward ω-continuity of $\cup$ is immediate. □

5.3.3 Approximating the distance of composed processes

Let (Σ, A, R) be any PGSOS PTSS. We compute in parallel the denotation of terms in $\mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)$ and of rules in R as least fixed point of a monotone function. We consider pairs of the form (τ, ρ), with $\tau : (\mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)) \to \mathscr{D}$ a mapping assigning to each open term $t \in \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)$ its denotation $\tau(t) \in \mathscr{D}$, and ρ a mapping assigning to each rule $r \in R$ its denotation $\rho(r) \in \mathscr{D}$.

Definition 5.34. The *denotational model* for a PTSS (Σ, A, R) is the structure $(S, \sqsubseteq)$, where $S = S_T \times S_R$ with

- $S_T = (\mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)) \to \mathscr{D}$;
- $S_R = R \to \mathscr{D}$,

for all $\tau, \tau' \in S_T$ and $\rho, \rho' \in S_R$ the ordering $(\tau, \rho) \sqsubseteq (\tau', \rho')$ iff $\forall t \in \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma).\tau(t) \sqsubseteq \tau'(t)$ and $\forall r \in R.\rho(r) \sqsubseteq \rho'(r)$.

We show first that $(S, \sqsubseteq)$ is a cpo with least element $(\bot_T, \bot_R)$ defined by $\bot_T(t) = \bot_R(r) = 0 \in \mathscr{D}$ for all terms $t \in \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)$ and rules $r \in R$.

Proposition 5.35. *$(S, \sqsubseteq)$ is a cpo with least element $(\bot_T, \bot_R)$.*

Proof. First, we show reflexivity. For any $(\tau, \rho) \in S$ we have $(\tau, \rho) \sqsubseteq (\tau, \rho)$ because $\tau(t) \sqsubseteq \tau(t)$ for all $t \in \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)$ and $\rho(r) \sqsubseteq \rho(r)$ for all $r \in R$ by reflexivity of $\sqsubseteq$ on $\mathscr{D}$ (Proposition 5.25).

In the same line of reasoning, we show transitivity. For any $(\tau_1, \rho_1), (\tau_2, \rho_2), (\tau_3, \rho_3) \in S$ with $(\tau_1, \rho_1) \sqsubseteq (\tau_2, \rho_2)$ and $(\tau_2, \rho_2) \sqsubseteq (\tau_3, \rho_3)$ we have $\tau_1(t) \sqsubseteq \tau_2(t) \sqsubseteq \tau_3(t)$ for all $t \in \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)$ and $\rho_1(r) \sqsubseteq \rho_2(r) \sqsubseteq \rho_3(r)$ for all $r \in R$. By transivity of $\sqsubseteq$ on $\mathscr{D}$ (Proposition 5.25) we get $\tau_1(t) \sqsubseteq \tau_3(t)$ and $\rho_1(r) \sqsubseteq \rho_3(r)$, thus implying $(\tau_1, \rho_1) \sqsubseteq (\tau_3, \rho_3)$.

We consider now antisymmetry. For any $(\tau_1, \rho_1), (\tau_2, \rho_2) \in S$ with $(\tau_1, \rho_1) \sqsubseteq (\tau_2, \rho_2)$ and $(\tau_2, \rho_2) \sqsubseteq (\tau_1, \rho_1)$ we have $\tau_1(t) \sqsubseteq \tau_2(t) \sqsubseteq \tau_1(t)$ for all $t \in \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)$ and $\rho_1(r) \sqsubseteq \rho_2(r) \sqsubseteq \rho_1(r)$ for all $r \in R$. By antisymmetry of $\sqsubseteq$ on $\mathscr{D}$ (Proposition 5.25) we

get $\tau_1(t) = \tau_2(t)$ for all $t \in \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)$ and $\rho_1(r) = \rho_2(r)$ for all $r \in R$, thus implying $(\tau_1, \rho_1) = (\tau_2, \rho_2)$.

We conclude with showing that any countable ascending chain has the supremum. Given a chain $(\tau_k, \rho_k)_{k\in\mathbb{N}}$ with $(\tau_k, \rho_k) \sqsubseteq (\tau_{k+1}, \rho_{k+1})$ for all $k \in \mathbb{N}$, by definition we have $\tau_k \sqsubseteq \tau_{k+1}$ and $\rho_k \sqsubseteq \rho_{k+1}$ for all $k \in \mathbb{N}$. Since all countable ascending chains on $\mathscr{D}$ have a supremum (Proposition 5.25), both chains $(\tau_k)_{k\in\mathbb{N}}$ and $(\rho_k)_{k\in\mathbb{N}}$ have a supremum. It is immediate to infer that $(\sup(\tau_k)_{k\in N}, \sup(\rho_k)_{k\in N})$ is the supremum for $(\tau_k, \rho_k)_{k\in\mathbb{N}}$. □

Given a rule $r \in R$, let X_r be the set of source variables x_i for which r tests the reactive behavior, i.e. $x_i \in X_r$ iff r has either some positive premise $x_i \xrightarrow{a_{i,m}} \mu_{i,m}$ or some negative premise $x_i \xrightarrow{b_{i,n}} \not\rightarrow$.

The mapping $\mathbf{F}\colon S \to S$ defined in Figure 5.1 computes iteratively the nondeterministic probabilistic multiplicities for all terms and rules. The denotation of a rule r is obtained from the denotation of its target $\mathrm{trgt}(r)$ by discounting the multiplicity of the source variables, and by mapping the multiplicity of the derivatives to is respective source variables (cf. Example 5.28). The denotation of a state term $f(t_1, \ldots, t_{r(f)})$ is defined as the union of the denotations of all rules in R_f combined with the denotation of the arguments, where the union over the denotations of the rules in R_f reflects the nondeterminism arising from the choice between those rules.

Example 5.36. Consider the open term $t = f(x)$ and the following rules r_1 and r_2:

$$\frac{x \xrightarrow{a} \mu}{f(x) \xrightarrow{a} \mu \parallel \mu} \qquad \frac{x \xrightarrow{a} \mu}{f(x) \xrightarrow{a} \delta(x) \parallel \delta(x) \parallel \delta(x)}$$

As in Example 5.28, we consider now the closed substitutions $\sigma_1(x) = a.a.a.0$ and $\sigma_2(x) = a.([0.9]a.a.0 \oplus [0.1]0)$ with $\mathbf{d}(\sigma_1(x), \sigma_2(x)) = 0.1\lambda$. The process distance induced by σ_1 and σ_2 is then $\mathbf{d}(\sigma_1, \sigma_2)(x) = 0.1\lambda$ and $\mathbf{d}(\sigma_1, \sigma_2)(\mu) = 0$. We get $\mathbf{d}(\sigma_1(t), \sigma_2(t)) = \max\{\lambda(1-(1-0.1)^2), \lambda^2(1-(1-0.1)^3)\}$.

By following Example 5.28 we get that the denotation of r_1 is $[\![r_1]\!] = \downarrow \{(2\lambda)_\mu \oplus 2_x\}$ and the denotation of r_2 is $[\![r_2]\!] = \downarrow \{(3\lambda)_x\}$, thus giving that the denotation of term t is $[\![t]\!] = \downarrow \{[\![r_1]\!] \odot_x [\![x]\!], [\![r_2]\!] \odot_x [\![x]\!]\} = \downarrow \{((2\lambda)_\mu \oplus 2_x) \odot_x 1_x, (3\lambda)_x \odot_x 1_x\} = \downarrow \{2_x, (3\lambda)_x\}$. Therefore, $\mathbf{A}([\![t]\!], \mathbf{d}(\sigma_1, \sigma_2)) = \sup\{\lambda(1-(1-0.1)^2), \lambda(1-(1-0.1)^{3\lambda})\} \geq \max\{\lambda(1-(1-0.1)^2), \lambda^2(1-(1-0.1)^3)\} = \mathbf{d}(\sigma_1(t), \sigma_2(t))$.

However, for distribution terms the application of the operator needs to consider two peculiarities, which are described by means of Examples 5.37 and 5.38 below.

First, in the distribution term $f(\theta_1, \ldots, \theta_{r(f)})$ the operator f may discriminate states in derivatives belonging to θ_i solely on the basis that in some rule $r \in R_f$ the argument x_i gets tested on the ability to perform or not perform some action.

Example 5.37. Consider the operators f and g defined by the following rules:

$$\frac{x \xrightarrow{a} \mu}{f(x) \xrightarrow{a} g(\mu)} \qquad \frac{y \xrightarrow{a} \nu}{g(y) \xrightarrow{a} \delta(0)}$$

Let us name with r_f and r_g the rules for f and g above. Operator f mimics the first move of its argument and then, by operator g, only tests the states in the derivative for their

Function $\mathbf{F}\colon S \to S$ is defined by $\mathbf{F}(\tau,\rho) = (\tau',\rho')$ with

$$\tau'(t) = \begin{cases} 1_x & \text{if } t = x \\ \bigoplus_{i=1}^{\mathsf{r}(f)} \big(\rho_f \odot_{x_i} \tau(t_i)\big) & \text{if } \left[\begin{array}{l} t = f(t_1,\ldots,t_{\mathsf{r}(f)}) \\ \rho_f = \bigcup_{r\in R_f} \rho(r) \end{array}\right. \end{cases}$$

$$\tau'(\theta) = \begin{cases} 1_\mu & \text{if } \theta = \mu \\ \tau(t) & \text{if } \theta = \delta(t) \\ \sum_{i\in I} q_i \cdot \tau(\theta_i) & \text{if } \theta = \sum_{i\in I} q_i\theta_i \\ \bigoplus_{i=1}^{\mathsf{r}(f)} \big(\rho_f \odot_{x_i} \tau(\theta_i)\big) & \text{if } \left[\begin{array}{l} \theta = f(\theta_1,\ldots,\theta_{\mathsf{r}(f)}) \\ \rho_f = \downarrow\left\{ \mathop{\mathbb{V}}_{r\in R_f} \rho_r \right\} \\ \rho_r = \left(\mathop{\mathbb{V}}_{p\in\rho(r)} p \right) \mathbb{V}\, 1_{X_r} \end{array}\right. \end{cases}$$

$$\rho'(r) = \left\{ (\lambda\cdot p) \oplus \Big(\bigoplus_{\substack{a_{i,m} \\ x_i \xrightarrow{\quad} \mu_{i,m} \in \\ \mathrm{pprem}(r)}} p \odot_{\mu_{i,m}} 1_{x_i} \Big) \mid p \in \tau(\mathrm{trgt}(r)) \right\}$$

Figure 5.1: Computation of the denotation of arbitrary terms

ability to perform action a. Consider first operator g. We get $\mathbf{d}(\sigma_1(g(y)),\sigma_2(g(y))) = 0$ for all closed substitutions σ_1,σ_2 such that $\mathbf{d}(\sigma_1(y),\sigma_2(y)) < 1$. Clearly, $[\![g(y)]\!] = 0$. Consider now $t = f(x)$ and substitutions $\sigma_1(x) = a.a.0$ and $\sigma_2(x) = a.([0.9]a.0 \oplus [0.1]0)$ with $\mathbf{d}(\sigma_1(x),\sigma_2(x)) = 0.1\lambda$. The distance between $\sigma_1(f(x))$ and $\sigma_2(f(x))$ depends on the distance between distributions $\delta(g(a.0))$ and $0.9\delta(g(a.0)) + 0.1\delta(g(0))$: from $\mathbf{d}(g(a.0),g(0)) = 1$ we get $\mathbf{d}(f(\sigma_1(x)),f(\sigma_2(x))) = \lambda\,\mathbf{K}(\mathbf{d})(\delta(g(a.0)), 0.9\delta(g(a.0)) + 0.1\delta(g(0))) = 0.1\lambda$.

If we would ignore that g tests its argument on the reactive behavior, then the denotation of $g(\mu)$ would be $[\![g(\mu)]\!] = [\![g(x)]\!] \odot_x 1_\mu = 0$, the denotation of the rule for f would be $[\![r_f]\!] = \lambda[\![g(\mu)]\!] \oplus ([\![g(\mu)]\!] \odot_\mu 1_x) = 0$, and the denotation for $f(x)$ would be $[\![f(x)]\!] = [\![r_f]\!] \odot_x 1_x = 0$. Note that this denotation does not approximate correctly from above the distance between $f(\sigma_1(x))$ and $f(\sigma_2(x))$ since $\mathbf{A}(0,\mathbf{d}(\sigma_1,\sigma_2)) = 0 < 0.1 = \mathbf{d}(f(\sigma_1(x)),f(\sigma_2(x)))$.

Because the operator g tests its argument on the ability to perform action a, it can discriminate instances of the derivative μ the same way as if the process would progress (without replication). Thus, the denotation of operator g if applied in the rule target is $\rho_g = \downarrow\{0 \mathbb{V}\, 1_{X_{r_g}}\} = 1_x$ as $X_{r_g} = \{x\}$. Hence, $[\![g(\mu)]\!] = \rho_g \odot_x 1_\mu = 1_\mu$. Thus, $[\![r_f]\!] = \lambda[\![g(\mu)]\!] \oplus ([\![g(\mu)]\!] \odot_\mu 1_x) = 1_x \oplus \lambda_\mu$, and $[\![f(x)]\!] = [\![r_f]\!] \odot_x 1_x = 1_x$. Notice that this denotation approximates correctly from above the distance between $f(\sigma_1(x))$ and

$f(\sigma_2(x))$ since $\mathbf{d}(f(\sigma_1(x)), f(\sigma_2(x))) = 0.1\lambda = \mathbf{A}([\![f(x)]\!], \mathbf{d}(\sigma_1, \sigma_2))$.

Second, different states in the support of a distribution term $f(\theta_1, \ldots, \theta_{\mathrm{r}(f)})$ may evolve according to different rules of R_f, which requires to combine the denotations of the distribution terms $\theta_1, \ldots, \theta_{\mathrm{r}(f)}$ with the supremum of the denotations of all rules in R_f instead of the union of the denotations of the rules in R_f.

Example 5.38. Consider the operator f defined by the following rule r_f:

$$\frac{x \xrightarrow{a} \mu}{f(x) \xrightarrow{a} \mu + \mu}$$

Operator f replicates the derivative μ of x and evolves as alternative composition of both process copies. Consider the closed substitutions $\sigma_1(x) = a.a.0$ and $\sigma_2(x) = a.([0.9]a.0 \oplus [0.1]b.0)$ with $\mathbf{d}(\sigma_1(x), \sigma_2(x)) = 0.1\lambda$. Then, $\mathbf{d}(\sigma_1(f(x)), \sigma_2(f(x))) = \lambda(1-(1-0.1)^2)$. The denotations for the two rules r_1 and r_2

$$r_1 = \frac{x_1 \xrightarrow{a} \mu_1}{x_1 + x_2 \xrightarrow{a} \mu_1} \qquad\qquad r_2 = \frac{x_2 \xrightarrow{a} \mu_2}{x_1 + x_2 \xrightarrow{a} \mu_2}$$

defining the alternative composition are the downward closed sets with maximal elements $(\lambda_{\mu_1} \oplus 1_{x_1}) \in \mathscr{P}$ and $(\lambda_{\mu_2} \oplus 1_{x_2}) \in \mathscr{P}$. Then, $\rho_{r_1} = (\lambda_{\mu_1} \oplus 1_{x_1}) \mathbin{\mathbb{V}} 1_{x_1} = (\lambda_{\mu_1} \oplus 1_{x_1}) \in \mathscr{P}$ and $\rho_{r_2} = (\lambda_{\mu_2} \oplus 1_{x_2}) \mathbin{\mathbb{V}} 1_{x_2} = (\lambda_{\mu_2} \oplus 1_{x_2}) \in \mathscr{P}$. Hence, the denotation of the operator alternative composition is $\rho_+ = \downarrow\{\rho_{r_1} \mathbin{\mathbb{V}} \rho_{r_2}\} = \lambda_{\{\mu_1,\mu_2\}} \oplus 1_{\{x_1,x_2\}} \in \mathscr{D}$. It follows that the denotation for the target of the f-defining rule is $[\![\mu + \mu]\!] = (\lambda_{\{\mu_1,\mu_2\}} \oplus 1_{\{x_1,x_2\}}) \odot_{x_1} 1_\mu) \oplus (\lambda_{\{\mu_1,\mu_2\}} \oplus 1_{\{x_1,x_2\}}) \odot_{x_2} 1_\mu) = 2_\mu$. Then, $[\![r_f]\!] = \lambda \cdot [\![\mu + \mu]\!] \oplus [\![\mu + \mu]\!] \odot_\mu 1_x = (2\lambda)_\mu + 2_x$. Thus, $[\![f(x)]\!] = [\![r_f]\!] \odot_x 1_x = 2_x$. Then, $\mathbf{d}(\sigma_1(f(x)), \sigma_2(f(x))) \leq \mathbf{A}(2_x, \mathbf{d}(\sigma_1, \sigma_2)) = \lambda(1-(1-0.1)^2)$.

To summarize Examples 5.37 and 5.38: The nondeterministic probabilistic multiplicity for operator f applied to some distribution term is given by $\rho_f = \downarrow\{\mathbb{V}_{r \in R_f} \rho_r\}$ with $\rho_r = (\mathbb{V}_{p \in \rho(r)} p) \mathbin{\mathbb{V}} 1_{X_r}$ (Figure 5.1). We explain this expression stepwise. For any rule r we define by $\mathbb{V}_{p \in \rho(r)} p \in \mathscr{P}$ the least probabilistic multiplicity which covers all nondeterministic choices represented by the probabilistic multiplicities in $\rho(r) \in \mathscr{D}$. By $\rho_r = (\mathbb{V}_{p \in \rho(r)} p) \mathbin{\mathbb{V}} 1_{X_r} \in \mathscr{P}$ we capture the case that premises of r only test source variables in X_r on their ability to perform an action (cf. Example 5.37). Then, by $\mathbb{V}_{r \in R_f} \rho_r \in \mathscr{P}$ we define the least probabilistic multiplicity which covers all choices of rules $r \in R_f$ (cf. Example 5.38). Finally, by the downward closure $\downarrow\{\mathbb{V}_{r \in R_f} \rho_r\} \in \mathscr{D}$ we gain the nondeterministic probabilistic multiplicity ρ_f that can be applied to the distribution term (Figure 5.1).

Proposition 5.39. $\mathbf{F}$ *is order-preserving and upward* ω*-continuous.*

Proof. We start with monotonicity. Assume $(\tau_1, \rho_1), (\tau_2, \rho_2) \in S$ with $(\tau_1, \rho_1) \sqsubseteq (\tau_2, \rho_2)$. Let $\mathbf{F}(\tau_1, \rho_1) = (\tau_1', \rho_1')$ and $\mathbf{F}(\tau_2, \rho_2) = (\tau_2', \rho_2')$. We need to show $(\tau_1', \rho_1') \sqsubseteq (\tau_2', \rho_2')$, namely $\tau_1'(t) \sqsubseteq \tau_2'(t)$ for all $t \in \mathbb{T}(\Sigma)$ and $\rho_1'(r) \sqsubseteq \rho_2'(r)$ for all $r \in R$.

First we show that $\tau_1'(t) \sqsubseteq \tau_2'(t)$ for all $t \in \mathbb{T}(\Sigma)$. We reason by structural induction over t. The base case $t = x$ is immediate since $\tau_1'(x) = 1_x = \tau_2'(x)$. Consider the inductive step $t = f(t_1, \ldots, t_{\mathrm{r}(f)})$. By definition $\tau_1'(f(t_1, \ldots, t_{\mathrm{r}(f)})) = \bigoplus_{i=1}^{\mathrm{r}(f)} (\rho_f^1 \odot_{x_i} \tau_1(t_i))$ with

$\rho_f^1 = \bigcup_{r\in R_f} \rho_1(r)$ and $\tau_2'(f(t_1,\dots,t_{\mathsf{r}(f)})) = \bigoplus_{i=1}^{\mathsf{r}(f)}(\rho_f^2 \odot_{x_i} \tau_2(t_i))$ with $\rho_f^2 = \bigcup_{r\in R_f} \rho_2(r)$. By $(\tau_1,\rho_1) \sqsubseteq (\tau_2,\rho_2)$ we get $\tau_1(t_i) \sqsubseteq \tau_2(t_i)$ for $i = 1,\dots,\mathsf{r}(f)$, and $\rho_1(r) \sqsubseteq \rho_2(r)$ for all $r \in R_f$. By monotonicity of $\cup$ on $\mathscr{D}$ (Proposition 5.32) we get $\rho_f^1 \sqsubseteq \rho_f^2$. By monotonicity of $\oplus$ and $\odot_{x_i}$ on $\mathscr{D}$ (Proposition 5.32) it follows $\tau_1'(f(t_1,\dots,t_{\mathsf{r}(f)})) \sqsubseteq \tau_2'(f(t_1,\dots,t_{\mathsf{r}(f)}))$.

We proceed by showing that $\tau_1'(\theta) \sqsubseteq \tau_2'(\theta)$ for all $\theta \in \mathbb{DT}(\Sigma)$. We reason by structural induction over θ. The base case $\theta = \mu$ is immediate since $\tau_1'(\mu) = 1_\mu = \tau_2'(\mu)$. The base case $\theta = \delta(t)$ is given by $\tau_1'(\theta) = \tau_1(t) \sqsubseteq \tau_2(t) = \tau_2'(\theta)$. Consider the inductive step $\theta = \sum_{i\in I} q_i\theta_i$. By definition $\tau_1'(\theta) = \sum_{i\in I} q_i\cdot\tau_1(\theta_i)$ and $\tau_2'(\theta) = \sum_{i\in I} q_i\cdot\tau_2(\theta_i)$. By $(\tau_1,\rho_1) \sqsubseteq (\tau_2,\rho_2)$ we have $\tau_1(\theta_i) \sqsubseteq \tau_2(\theta_i)$ for all $i \in I$. Hence, by monotonicity of $+$ on $\mathscr{D}$ (Proposition 5.32) it follows $\tau_1'(\theta) = \sum_{i\in I} q_i \cdot \tau_1(\theta_i) \sqsubseteq \sum_{i\in I} q_i \cdot \tau_2(\theta_i) = \tau_2'(\theta)$. Finally, consider the inductive step $\theta = f(\theta_1,\dots,\theta_{\mathsf{r}(f)})$. By definition $\tau_1'(f(\theta_1,\dots,\theta_{\mathsf{r}(f)})) = \bigoplus_{i=1}^{\mathsf{r}(f)}(\rho_f^1 \odot_{x_i} \tau_1(\theta_i))$ with $\rho_f^1 = \downarrow \{\mathbb{V}_{r\in R_f}\rho_r^1\}$ and $\rho_r^1 = (\mathbb{V}_{p\in\rho_1(r)}p) \mathbb{V} 1_{X_r}$, and $\tau_2'(f(\theta_1,\dots,\theta_{\mathsf{r}(f)})) = \bigoplus_{i=1}^{\mathsf{r}(f)}(\rho_f^2 \odot_{x_i} \tau_2(\theta_i))$ with $\rho_f^2 = \downarrow \{\mathbb{V}_{r\in R_f}\rho_r^2\}$ and $\rho_r^2 = (\mathbb{V}_{p\in\rho_2(r)}p) \mathbb{V} 1_{X_r}$. By $(\tau_1,\rho_1) \sqsubseteq (\tau_2,\rho_2)$ we have $\tau_1(\theta_i) \sqsubseteq \tau_2(\theta_i)$ for $i = 1,\dots,\mathsf{r}(f)$, and $\rho_1(r) \sqsubseteq \rho_2(r)$ for each $r \in R_f$. By monotonicity of $\mathbb{V}$ on $\mathscr{D}$ (Proposition 5.32) we get $\rho_r^1 \sqsubseteq \rho_r^2$ and, then, $\rho_f^1 \sqsubseteq \rho_f^2$. Then, by monotonicity of $\oplus$ and $\odot_{x_i}$ on $\mathscr{D}$ (Proposition 5.32) it follows $\tau_1'(f(\theta_1,\dots,\theta_{\mathsf{r}(f)})) \sqsubseteq \tau_2'(f(\theta_1,\dots,\theta_{\mathsf{r}(f)}))$.

To conclude the proof of monotonicity we have to show that $\rho_1'(r) \sqsubseteq \rho_2'(r)$ for all $r \in R$. By definition for $r \in R$ we have $\rho_1'(r) = \{(\lambda\cdot p) \oplus (\bigoplus_{x_i \xrightarrow[\mathsf{pprem}(r)]{a_{i,m}} \mu_{i,m}\in} p \odot_{\mu_{i,m}} 1_{x_i}) \mid p \in \tau_1(\mathsf{trgt}(r))\}$ and $\rho_2'(r) = \{(\lambda\cdot p) \oplus (\bigoplus_{x_i \xrightarrow[\mathsf{pprem}(r)]{a_{i,m}} \mu_{i,m}\in} p \odot_{\mu_{i,m}} 1_{x_i}) \mid p \in \tau_2(\mathsf{trgt}(r))\}$. By $(\tau_1,\rho_1) \sqsubseteq (\tau_2,\rho_2)$ we get $\tau_1(\mathsf{trgt}(r)) \sqsubseteq \tau_2(\mathsf{trgt}(r))$ for each $r \in R$. By monotonicity of $\oplus$, $\odot_{x_i}$ and $\cdot$ on $\mathscr{D}$ (Proposition 5.32) it follows $\rho_1'(r) \sqsubseteq \rho_2'(r)$.

To prove upward ω-continuity we show $\mathbf{F}(\sup(\tau_k,\rho_k)_{k\in\mathbb{N}}) = \sup(\mathbf{F}(\tau_k,\rho_k))_{k\in\mathbb{N}}$ for all ω-chains $(\tau_k,\rho_k)_{k\in\mathbb{N}}$. Let $(\tau_1',\rho_1') = \mathbf{F}(\sup(\tau_k,\rho_k)_{k\in\mathbb{N}})$ and $(\tau_2',\rho_2') = \sup(\mathbf{F}(\tau_k,\rho_k))_{k\in\mathbb{N}}$. To prove $(\tau_1',\rho_1') = (\tau_2',\rho_2')$ we have to prove that $\tau_1'(t) = \tau_2'(t)$ for all $t \in \mathbb{T}(\Sigma)\cup\mathbb{DT}(\Sigma)$ and $\rho_1'(r) = \rho_2'(r)$ for all $r \in R$.

First we show that $\tau_1'(t) = \tau_2'(t)$ for all $t \in \mathbb{T}(\Sigma)$. We reason by structural induction over t. The base case $t = x$ is immediate since $\tau_1'(x) = 1_x = \tau_2'(x)$. Consider the inductive step $t = f(t_1,\dots t_{\mathsf{r}(f)})$. We get $\tau_1'(f(t_1,\dots,t_{\mathsf{r}(f)})) = \bigoplus_{i=1}^{\mathsf{r}(f)}(\bigcup_{r\in R_f} \sup(\rho_k(r))_{k\in\mathbb{N}} \odot_{x_i} \sup(\tau_k(t_i))_{k\in\mathbb{N}})$, $\tau_2'(f(t_1,\dots,t_{\mathsf{r}(f)})) = \sup(\bigoplus_{i=1}^{\mathsf{r}(f)}(\bigcup_{r\in R_f} \rho_k(r) \odot_{x_i} \tau_k(t_i))_{k\in\mathbb{N}}$. Then, upward ω-continuity of operations $\cup, \odot_{x_i}, \oplus$ (Proposition 5.33) gives $\tau_1'(f(t_1,\dots,t_{\mathsf{r}(f)})) = \tau_2'(f(t_1,\dots,t_{\mathsf{r}(f)}))$.

We proceed by showing that $\tau_1'(\theta) = \tau_2'(\theta)$ for all $\theta \in \mathbb{DT}(\Sigma)$. We reason by structural induction over θ. The base case $\theta = \mu$ is immediate since $\tau_1'(\mu) = 1_\mu = \tau_2'(\mu)$. Consider the base case $\theta = \delta_t$. By definition we have $\tau_1'(\theta) = \sup((\tau_k(t))_{k\in\mathbb{N}}) = \tau_2'(\theta)$ and the thesis follows immediately. Consider the inductive step $\theta = \sum_{i\in I} q_i\theta_i$. By definition we have $\tau_1'(\sum_{i\in I} q_i\theta_i) = \sum_{i\in I} q_i \cdot \sup(\tau_k(\theta_i))_{k\in\mathbb{N}}$ and $\tau_2'(\sum_{i\in I} q_i\theta_i) = \sup(\sum_{i\in I} q_i \cdot \tau_k(\theta_i))_{k\in\mathbb{N}}$. By upward ω-continuity of $+$ and $\cdot$ (Proposition 5.33) we get $\tau_1'(\sum_{i\in I} q_i\theta_i) = \tau_2'(\sum_{i\in I} q_i\theta_i)$. Finally, consider the inductive step $\theta = f(\theta_1,\dots,\theta_{\mathsf{r}(f)})$. Then we have $\tau_1'(f(\theta_1,\dots,\theta_{\mathsf{r}(f)})) = \bigoplus_{i=1}^{\mathsf{r}(f)}(\downarrow \{\mathbb{V}_{r\in R_f}((\mathbb{V}_{p\in\sup(\rho_k)_{k\in\mathbb{N}}(r)}p) \mathbb{V} 1_{X_r})\} \odot_{x_i} \sup(\tau_k)_{k\in\mathbb{N}}(\theta_i))$ and $\tau_2'(f(\theta_1,\dots,\theta_{\mathsf{r}(f)})) = \sup(\bigoplus_{i=1}^{\mathsf{r}(f)}(\downarrow \{\mathbb{V}_{r\in R_f}((\mathbb{V}_{p\in\rho_k)(r)}p) \mathbb{V} 1_{X_r})\} \odot_{x_i} \tau_k(\theta_i)))_{k\in\mathbb{N}}$. By upward

ω-continuity of $\oplus$, $\odot_{x_i}$ and $\between$ (Proposition 5.33) it follows that $\tau_1'(f(\theta_1,\dots,\theta_{\mathsf{r}(f)})) = \tau_2'(f(\theta_1,\dots,\theta_{\mathsf{r}(f)}))$.

To conclude we have to show that $\rho_1'(r) = \rho_2'(r)$ for all $r \in R$. By definition for $r \in R$ we have $\rho_1'(r) = \{(\lambda \cdot p) \oplus (\bigoplus_{x_i \xrightarrow[\mathsf{pprem}(r)]{a_{i,m}} \mu_{i,m} \in} p \odot_{\mu_{i,m}} 1_{x_i}) \mid p \in \sup(\tau_k)_{k\in\mathbb{N}}(\mathsf{trgt}(r))\}$ and $\rho_2'(r) = \sup(\{(\lambda \cdot p) \oplus (\bigoplus_{x_i \xrightarrow[\mathsf{pprem}(r)]{a_{i,m}} \mu_{i,m} \in} p \odot_{\mu_{i,m}} 1_{x_i}) \mid p \in \tau_k(\mathsf{trgt}(r))\})_{k\in\mathbb{N}}$. By upward ω-continuity of $\oplus$, $\between$ and $\cdot$ (Proposition 5.33) it follows $\rho_1'(r) = \rho_2'(r)$. □

Since $\mathbf{F}$ is monotone and upwardly ω-continuous, $\mathbf{F}$ has a least fixed point.

Proposition 5.40. $\sup(\mathbf{F}^n(\bot_T, \bot_R))_{n\in\mathbb{N}}$ *is the least fixed point of* $\mathbf{F}$.

Proof. Directly by Knaster-Tarski fixed point theorem, since $(S, \sqsubseteq)$ is a cpo (Proposition 5.35) and $\mathbf{F}$ is monotone and upwardly ω-continuous (Proposition 5.39). □

We denote by (ω_T, ω_R) the least fixed point of $\mathbf{F}$. We write $[\![t]\!]$ for $\omega_T(t)$. We call $[\![t]\!]$ the *canonical denotation* of t. It is not hard to verify that all denotations presented in Section 5.2 for P_{PA} are canonical. For any $\tau \in S_T$, we will write $[\![t]\!]_\tau$ for $\tau(t)$.

Definition 5.41. A denotation of terms $\tau \in S_T$ is *consistent* with a distance function $d \in [0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)}$, notation $d \preceq [\![\cdot]\!]_\tau$, if $d(\sigma_1(t), \sigma_2(t)) \leq \mathbf{A}([\![t]\!]_\tau, d(\sigma_1, \sigma_2))$ for all terms $t \in \mathbb{T}(\Sigma)$ and all closed substitutions σ_1, σ_2.

Now we can show that the functional $\mathbf{B}$ to compute the bisimulation distance and functional $\mathbf{F}$ to compute the denotations preserve consistency (Proposition 5.44). A simple inductive argument allows then to show that the canonical denotation of terms $[\![\cdot]\!]$ is consistent with the bisimilarity metric $\mathbf{d}$ (Theorem 5.45).

First we need to show two auxiliary properties of $\mathbf{K}$ and $\mathbf{P}$. The following Lemma shows how the convex combination distributes over the Kantorovich functional $\mathbf{K}$ and the probabilistic distance approximation functional $\mathbf{P}$.

Lemma 5.42. *Let* $d \in [0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)}$, σ, σ' *be closed substitutions,* $\sum_{i\in I} q_i\theta_i \in \mathbb{DT}(\Sigma)$ *a distribution term and* $\sum_{i\in I} q_i p_i \in \mathscr{P}$ *a probabilistic multiplicity. Then:*

1. $\mathbf{K}(d)\left(\sum_{i\in I} q_i \cdot \sigma(\theta_i), \sum_{i\in I} q_i \cdot \sigma'(\theta_i)\right) \quad \leq \quad \sum_{i\in I} q_i \cdot \mathbf{K}(d)(\sigma(\theta_i), \sigma'(\theta_i))$;
2. $\mathbf{P}(\sum_{i\in I} q_i \cdot p_i, e) = \sum_{i\in I} q_i \cdot \mathbf{P}(p_i, e)$.

Proof. Lemma 5.42.1 follows directly from Proposition 2.32.3. Lemma 5.42.2 follows directly from Lemma 5.31.5. □

Given two closed instances of a composed distribution term, the independent product of matchings of the subterms is a matching between the closed instances of the composed distribution term.

Lemma 5.43. *Let* σ, σ' *be closed substitutions and* $\theta = f(\theta_1,\dots,\theta_{\mathsf{r}(f)}) \in \mathbb{DT}(\Sigma)$. *Then*

$$\mathbf{K}(d)(\sigma(\theta), \sigma'(\theta)) \leq \sum_{t,t'\in\mathsf{T}(\Sigma)} d(t,t') \cdot \omega(t,t')$$

with $\omega(t,t') = \prod_{i=1}^{\mathsf{r}(f)} \omega_i(t_i, t_i')$, *if* $t = f(t_1,\dots,t_{\mathsf{r}(f)})$ *and* $t' = f(t_1',\dots,t_{\mathsf{r}(f)}')$, *and* $\omega(t,t') = 0$ *otherwise, and* ω_i *is defined such that* $\mathbf{K}(d)(\sigma(\theta_i), \sigma'(\theta_i)) = \sum_{t_i,t_i'\in\mathsf{T}(\Sigma)} d(t_i,t_i')\cdot\omega_i(t_i,t_i')$.

Proof. It suffices to show that $\omega \in \Omega(\sigma(\theta), \sigma'(\theta))$. It is clear that $\omega \in \Delta(\mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma))$ because it is defined as joined density of distributions $\omega_i \in \Delta(\mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma))$. The marginal distribution of ω with respect to $\sigma(\theta)$ is:

$$
\begin{aligned}
&\sum_{t' \in \mathsf{T}(\Sigma)} \omega(f(t_1, \ldots, t_{\mathsf{r}(f)}), t') \\
=& \sum_{f(t'_1, \ldots, t'_{\mathbf{r}(f)}) \in \mathsf{T}(\Sigma)} \omega(f(t_1, \ldots, t_{\mathsf{r}(f)}), f(t'_1, \ldots, t'_{\mathsf{r}(f)})) \\
=& \sum_{\substack{t'_i \in \mathbf{T}(\Sigma) \\ \text{for } i=1,\ldots,\mathbf{r}(f)}} \prod_{i=1}^{\mathsf{r}(f)} \omega_i(t_i, t'_i) \\
=& \prod_{i=1}^{\mathsf{r}(f)} \sum_{t'_i \in \mathsf{T}(\Sigma)} \omega_i(t_i, t'_i) \\
=& \prod_{i=1}^{\mathsf{r}(f)} \sigma(\theta_i)(t_i) \qquad \Big(\text{by } \sum_{t'_i \in \mathsf{T}(\Sigma)} \omega_i(t_i, t'_i) = \sigma(\theta_i)(t_i)\Big) \\
=& \sigma(\theta)(f(t_1, \ldots, t_{\mathsf{r}(f)}))
\end{aligned}
$$

and $\sum_{t' \in \mathsf{T}(\Sigma)} \omega(t, t') = 0 = \sigma(\theta)(t)$ if t is not in the form $t = f(t_1, \ldots, t_{\mathsf{r}(f)})$. The marginal distribution of ω with respect to $\sigma'(\theta)$ is calculated in an analogous manner. We conclude, ω is a matching for $\sigma(\theta)$ and $\sigma'(\theta)$). □

Lemmas 5.29 and 5.30 allow us now to prove that the functional **B** to compute the bisimulation distance and the functional **F** to compute the denotation of terms preserve consistency.

Proposition 5.44. *Let $d \in [0,1]^{\mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma)}$ with $d \sqsubseteq \mathbf{B}(d) = d'$ and $(\tau, \rho) \in S$ with $(\tau, \rho) \sqsubseteq \mathbf{F}(\tau, \rho) = (\tau', \rho')$. If $d \preceq [\![\cdot]\!]_\tau$ then $d' \preceq [\![\cdot]\!]_{\tau'}$.*

Proof. Let $\underline{\sigma}_1, \underline{\sigma}_2$ be closed substitutions and $f \in \Sigma$. Define $t = f(x_1, \ldots, x_{\mathsf{r}(f)})$, $t_1 = \underline{\sigma}_1(t)$ and $t_2 = \underline{\sigma}_2(t)$. Let $e' = d'(\underline{\sigma}_1, \underline{\sigma}_2)$. We will show $d'(t_1, t_2) \leq \mathbf{A}([\![t]\!]_{\tau'}, e')$ by exploiting that $d \preceq [\![\cdot]\!]_\tau$ is given, i.e. $d(s, s') \leq \mathbf{A}([\![g(x_1, \ldots, x_{\mathsf{r}(g)})]\!], e)$ for all closed substitutions σ, σ', $g \in \Sigma$, $s = \sigma(g(x_1, \ldots, x_{\mathsf{r}(g)}))$, $s' = \sigma'(g(x_1, \ldots, x_{\mathsf{r}(g)}))$, $e = d(\sigma, \sigma')$. If $e'(x_i) = 1$ for some argument variable x_i with $i = 1, \ldots, \mathsf{r}(f)$, then $d'(t_1, t_2) \leq \mathbf{A}([\![t]\!]_{\tau'}, e')$ is immediate since $\mathbf{A}([\![t]\!]_{\tau'}, e') = 1$. We consider the case $e'(x_i) < 1$ for all argument variables x_i.

By definition of **B** it suffices to show that if $t_1 \xrightarrow{a} \pi_1$ for some distribution $\pi_1 \in \Delta(\mathsf{T}(\Sigma))$ and action $a \in A$, then there exists a transition $t_2 \xrightarrow{a} \pi_2$ for a distribution $\pi_2 \in \Delta(\mathsf{T}(\Sigma))$ such that $\lambda \cdot \mathbf{K}(d)(\pi_1, \pi_2) \leq \mathbf{A}([\![t]\!]_{\tau'}, e')$. We will show that the transition $t_2 \xrightarrow{a} \pi_2$ is inferred by the same PGSOS rule used to infer the transition $t_1 \xrightarrow{a} \pi_1$. The transition $t_1 \xrightarrow{a} \pi_1$ is derived from a PGSOS-rule $r \in R_f$ given by

$$
\frac{\{x_i \xrightarrow{a_{i,m}} \mu_{i,m} \mid i \in I, m \in M_i\} \qquad \{x_i \xrightarrow{b_{i,n}} \not\to \mid i \in I, n \in N_i\}}{f(x_1, \ldots, x_{\mathsf{r}(f)}) \xrightarrow{a} \theta}
$$

with the substitution σ_1 such that $\sigma_1(x_i) = \underline{\sigma}_1(x_i)$ for $i = 1, \ldots, \mathsf{r}(f)$. Notice that $\sigma_1(\theta) = \pi_1$.

In the remainder we will first give suitable moves $\underline{\sigma}_2(x_i) \xrightarrow{a_{i,m}} \pi^2_{i,m}$ for all $i \in I$ and $m \in M_i$. This allows us to define the substitution σ_2 for all x_i and $\mu_{i,m}$ as $\sigma_2(x_i) = \underline{\sigma}_2(x_i)$ and $\sigma_2(\mu_{i,m}) = \pi^2_{i,m}$. Subsequently we show that $\sigma_2(x_i) \xrightarrow{b_{i,n}}\!\!\!\!\!\!/\;$ for all $i \in I$ and $n \in N_i$, thus inferring that the transition $t_2 \xrightarrow{a} \sigma_2(\theta)$ can be derived from rule r with substitution σ_2. Finally we show

$$\lambda \cdot \mathbf{K}(d)(\sigma_1(\theta), \sigma_2(\theta)) \leq \mathbf{A}([\![t]\!]_{\tau'}, e') \tag{5.11}$$

which confirms that $\sigma_2(\theta)$ is the distribution π_2 we were looking for. In detail, we show the stricter statement

$$\lambda \cdot \mathbf{K}(d)(\sigma_1(\theta), \sigma_2(\theta)) \leq \sup_{p \in P} \mathbf{P}(p, e'), \text{ with } P = \bigoplus_{i=1}^{\mathsf{r}(f)} (\rho(r) \odot_{x_i} \tau(x_i)). \tag{5.12}$$

Notice that Equation 5.12 is stricter than Equation 5.11 from the definition $\mathbf{A}([\![t]\!]_{\tau'}, e') = \sup_{p \in [\![t]\!]_{\tau'}} \mathbf{P}(p, e')$ and the relation $[\![t]\!]_{\tau'} = \bigoplus_{i=1}^{\mathsf{r}(f)}((\bigcup_{r' \in R_f} \rho(r')) \odot_{x_i} \tau(x_i)) \supseteq P$.

Consider the positive premises $x_i \xrightarrow{a_{i,m}} \mu_{i,m} \in \mathrm{pprem}(r)$ of rule r. Because $e'(x_i) = d'(\sigma_1(x_i), \underline{\sigma}_2(x_i)) < 1$ and $d' = \mathbf{B}(d)$, we get by Proposition 2.30 that $\underline{\sigma}_2(x_i)$ can perform an $a_{i,m}$ move. Moreover, by definition of $\mathbf{B}$ there is a transition $\underline{\sigma}_2(x_i) \xrightarrow{a_{i,m}} \pi^2_{i,m}$ for a distribution $\pi^2_{i,m}$ satisfying $\lambda \cdot \mathbf{K}(d)(\sigma_1(\mu_{i,m}), \pi^2_{i,m}) \leq e'(x_i)$. Define $\sigma_2(\mu_{i,m}) = \pi^2_{i,m}$.

Consider the negative premises $x_i \xrightarrow{b_{i,n}}\!\!\!\!\!\!/\;\; \in \mathrm{nprem}(r)$. Since $e'(x_i) = d'(\sigma_1(x_i), \underline{\sigma}_2(x_i)) < 1$ and $d' = \mathbf{B}(d)$, we get by Proposition 2.30 that $\underline{\sigma}_2(x_i) \xrightarrow{b_{i,n}}\!\!\!\!\!\!/\;$.

To summarize, all premises of $\sigma_2(r)$ are satisfied. Hence, by applying r the transition $t_2 \xrightarrow{a} \sigma_2(\theta)$ can be derived. It remains to show Equation 5.12. We proceed by structural induction over the rule target θ.

Consider the base case $\theta = \mu \in \mathscr{V}_d$. The PGSOS format ensures that $\mu = \mu_{j,m}$ for some $j \in I$ and $m \in M_j$. Then $\tau(\theta) = \{1_{\mu_{j,m}}\}$ and $\rho(r) = \{(\lambda \cdot p) \oplus (\bigoplus_{x_i \xrightarrow[\mathrm{pprem}(r)]{a_{i,m}} \mu_{i,m} \in} p \odot_{\mu_{i,m}} 1_{x_i}) \mid p \in \tau(\theta)\} = \{(\lambda \cdot 1_{\mu_{j,m}}) \oplus (\bigoplus_{x_i \xrightarrow[\mathrm{pprem}(r)]{a_{i,m}} \mu_{i,m} \in} 1_{\mu_{j,m}} \odot_{\mu_{i,m}} 1_{x_i})\} = \{\lambda_{\mu_{j,m}} \oplus 1_{x_j}\}$. Hence, the set P in equation 5.12 becomes $P = \bigoplus_{i=1}^{\mathsf{r}(f)}(\rho(r) \odot_{x_i} 1_{x_i}) = \bigoplus_{i=1}^{\mathsf{r}(f)}((\lambda_{\mu_{j,m}} \oplus 1_{x_j}) \odot_{x_i} 1_{x_i}) = \{1_{x_j}\}$. This reduces the right-hand side of Equation 5.12 to $\sup_{p \in P} \mathbf{P}(p, e') = \mathbf{P}(1_{x_j}, e') = d'(\sigma_1(x_j), \sigma_2(x_j)) = e'(x_j)$. Hence Equation 5.12 becomes $\lambda \cdot \mathbf{K}(d)(\sigma_1(\mu_{j,m}), \sigma_2(\mu_{j,m})) \leq e'(x_j)$, which holds by construction of σ_1 and σ_2 (see positive premise case above).

Consider the base case $\theta = \delta_t$. Then we have $\tau(\theta) = \tau(\delta_t) = \tau(t)$. Now we get $\rho(r) = \{(\lambda \cdot p) \oplus (\bigoplus_{x_i \xrightarrow[\mathrm{pprem}(r)]{a_{i,m}} \mu_{i,m} \in} p \odot_{\mu_{i,m}} 1_{x_i}) \mid p \in \tau(t)\} = \{\lambda \cdot p \mid p \in \tau(t)\}$ because for all $p \in \tau(t)$ we have $p(m) = 0$ for all multiplicities $m \in \mathscr{M}$ such that $m(\mu_{i,m}) > 0$ for any distribution variable $\mu_{i,m} \in \mathscr{V}_d$. Hence, the set P in equation 5.12 becomes $P = \bigoplus_{i=1}^{\mathsf{r}(f)}(\rho(r) \odot_{x_i} 1_{x_i}) = \bigoplus_{i=1}^{\mathsf{r}(f)}(\{\lambda \cdot p \mid p \in \tau(t)\} \odot_{x_i} 1_{x_i}) = \{\lambda \cdot p \mid p \in \tau(t)\}$. This reduces the right-hand side of Equation 5.12 to $\sup_{p \in P} \mathbf{P}(p, e') = \sup_{p \in \{\lambda \cdot p \mid p \in \tau(t)\}} \mathbf{P}(p, e') = \lambda \sup_{p \in [\![t]\!]_\tau} \mathbf{P}(p, e') = \lambda \cdot \mathbf{A}([\![t]\!]_\tau, e')$. Then, the left-hand side of Equation 5.12 satisfies

$\lambda \cdot \mathbf{K}(d)(\sigma_1(\theta), \sigma_2(\theta)) \leq \lambda \cdot d(\sigma_1(t), \sigma_2(t))$ by Proposition 2.32.2. Hence Equation 5.12 follows from $\lambda \cdot d(\sigma_1(t), \sigma_2(t)) \leq \lambda \cdot \mathbf{A}([\![t]\!]_\tau, e')$, namely $d(\sigma_1(t), \sigma_2(t)) \leq \mathbf{A}([\![t]\!]_\tau, e')$. By the hypothesis $d \preceq [\![\cdot]\!]_\tau$ we have $d(\sigma_1(t), \sigma_2(t)) \leq \mathbf{A}([\![t]\!]_\tau, e)$. By $e \sqsubseteq e'$ and monotonicity of $\mathbf{A}$ (Proposition 5.26) we get $\mathbf{A}([\![t]\!]_\tau, e) \leq \mathbf{A}([\![t]\!]_\tau, e')$. Hence, equation 5.12 holds.

For the inductive steps we introduce the operator src: $\mathscr{D} \times R \to \mathscr{D}$ that maps nondeterministic probabilistic multiplicities derived from the target of a rule r (and hence defining nondeterministic probabilistic multiplicities of source and derivative variables of r) to nondeterministic probabilistic multiplicities over only the source variables of r. The positive premises of r specify the relation between the source variables and derivative variables of r. The operator src is defined by:

$$\mathrm{src}(P, r) = \left\{ \bigoplus_{i=1}^{\mathsf{r}(f)} \left(\left((\lambda \cdot p) \oplus \bigoplus_{x_i \xrightarrow{q_{i,m}} \mu_{i,m} \in \mathrm{pprem}(r)} p \odot_{\mu_{i,m}} 1_{x_i} \right) \odot_{x_i} 1_{x_i} \right) \mid p \in P \right\}.$$

It is not hard to show that

$$\mathrm{src}\left(\sum_{i \in I} q_i \cdot P_i, r \right) = \sum_{i \in I} q_i \cdot \mathrm{src}(P_i, r) \tag{5.13}$$

$$\mathrm{src}\left(\bigoplus_{i \in I} P_i, r \right) = \bigoplus_{i \in I} \mathrm{src}(P_i, r) \tag{5.14}$$

$$\mathrm{src}\left(\bigoplus_{i \in I} (P \odot_{x_i} P_i), r \right) = \bigoplus_{i \in I} \left(P \odot_{x_i} \mathrm{src}(P_i, r) \right) \tag{5.15}$$

by exploiting Lemma 5.29.

Moreover, by

$$\bigoplus_{i=1}^{\mathsf{r}(f)} (\rho(r) \odot_{x_i} \tau(x_i)) = \mathrm{src}(\tau(\mathrm{trgt}(r)), r)$$

and by $\mathrm{trgt}(r) = \theta$ the proof obligation in Equation 5.12 can reformulated to

$$\lambda \cdot \mathbf{K}(d)(\sigma_1(\theta), \sigma_2(\theta)) \leq \sup_{p \in P} \mathbf{P}(p, e'), \text{ with } P = \mathrm{src}(\tau(\theta), r). \tag{5.16}$$

Equation 5.16 can be replaced by the stricter statement

$$\lambda \cdot \mathbf{K}(d)(\sigma_1(\theta), \sigma_2(\theta)) \leq \mathbf{P}(\hat{p}, e') \text{ for some } \hat{p} \in \mathrm{src}(\tau(\theta), r). \tag{5.17}$$

For the following structural induction over θ, Equation 5.17 is both the induction hypothesis for the subterms and the proof obligation for the structurally constructed term out of the subterms.

Consider $\theta = \sum_{i \in I} q_i \cdot \theta_i$. By Lemma 5.42.1 we have $\lambda \cdot \mathbf{K}(d)(\sigma_1(\theta), \sigma_2(\theta)) \leq \lambda \cdot \sum_{i \in I} q_i \cdot \mathbf{K}(d)(\sigma_1(\theta_i), \sigma_2(\theta_i))$. By the induction hypothesis (Equation 5.17) we obtain that $\lambda \cdot \sum_{i \in I} q_i \cdot \mathbf{K}(d)(\sigma_1(\theta_i), \sigma_2(\theta_i)) \leq \sum_{i \in I} q_i \cdot \mathbf{P}(\hat{p}_i, e')$ for some $\hat{p}_i \in \mathrm{src}(\tau(\theta_i), r)$. Since operator $+$ distributes over $\mathbf{P}$ (Lemma 5.31.5) we derive $\sum_{i \in I} q_i \cdot \mathbf{P}(\hat{p}_i, e') = \mathbf{P}(\sum_{i \in I} q_i \cdot$

$\hat{p}_i, e')$. Define $\hat{p} = \sum_{i\in I} q_i \cdot \hat{p}_i$. By Equation 5.13, $\hat{p}_i \in \mathrm{src}(\tau(\theta_i), r)$ allow us to infer that $\hat{p} \in \mathrm{src}(\tau(\sum_{i\in I} q_i \cdot \theta_i), r) = \mathrm{src}(\tau(\theta), r)$. Summarizing, $\lambda \cdot \mathbf{K}(d)(\sigma_1(\theta), \sigma_2(\theta)) \leq \mathbf{P}(\hat{p}, e')$ for some $\hat{p} \in \mathrm{src}(\tau(\theta), r)$, which confirms that the thesis holds.

Consider $\theta = g(\theta_1, \ldots, \theta_{\mathsf{r}(g)})$. By Definition of $\mathbf{K}$, for all $i = 1, \ldots, \mathsf{r}(g)$ we have

$$\mathbf{K}(d)(\sigma_1(\theta_i), \sigma_2(\theta_i)) = \sum_{u_i, v_i \in \mathsf{T}(\Sigma)} \omega_i(u_i, v_i) \cdot d(u_i, v_i)$$

for some matching $\omega_i \in \Omega(\sigma_1(\theta_i), \sigma_2(\theta_i))$. By Lemma 5.43 we get that the distribution $\omega \in \Delta(\mathsf{T}(\Sigma) \times \mathsf{T}(\Sigma))$ defined by $\omega(g(u_1, \ldots, u_{\mathsf{r}(g)}), g(v_1, \ldots, v_{\mathsf{r}(g)})) = \prod_{i=1}^{\mathsf{r}(g)} \omega_i(u_i, v_i)$ is a matching $\omega \in \Omega(\sigma_1(\theta), \sigma_2(\theta))$ such that

$$\mathbf{K}(d)(\sigma_1(\theta), \sigma_2(\theta)) \leq \sum_{\substack{u = g(u_1, \ldots, u_{\mathsf{r}(g)}) \\ v = g(v_1, \ldots, v_{\mathsf{r}(g)})} \in \mathsf{T}(\Sigma)} \left(\prod_{i=1}^{\mathsf{r}(g)} \omega_i(u_i, v_i) \right) \cdot d(u, v)$$

namely

$$\lambda \cdot \mathbf{K}(d)(\sigma_1(\theta), \sigma_2(\theta)) \leq \lambda \cdot \sum_{\substack{u = g(u_1, \ldots, u_{\mathsf{r}(g)}) \\ v = g(v_1, \ldots, v_{\mathsf{r}(g)})} \in \mathsf{T}(\Sigma)} \left(\prod_{i=1}^{\mathsf{r}(g)} \omega_i(u_i, v_i) \right) \cdot d(u, v). \tag{5.18}$$

Remember that for each rule $r \in R_g$ the indices of source variables for which r tests the reactive behavior is denoted by X_r. This lifts to operator g by $X_g = \bigcup_{r \in R_g} X_r$.
By the hypothesis $d \preceq [\![\cdot]\!]_\tau$ the distance between terms $u = g(u_1, \ldots, u_{\mathsf{r}(g)})$ and $v = g(v_1, \ldots, v_{\mathsf{r}(g)})$ expressed by d is approximated from above by

$$d(g(u_1, \ldots, u_{\mathsf{r}(g)}), g(v_1, \ldots, v_{\mathsf{r}(g)})) \leq \mathbf{A}([\![g(x_1, \ldots, x_{\mathsf{r}(g)})]\!]_\tau, e_{u,v})$$

with $e_{u,v} \in \mathscr{E}$ the process distance defined by $e_{u,v}(x_i) = d(u_i, v_i)$.
Moreover, if $d(u_i, v_i) < 1$ for all $1 \leq i \leq \mathsf{r}(g)$ then by the definition of $\tau(g(x_1, \ldots, x_{\mathsf{r}(g)}))$ we have

$$\mathbf{A}([\![g(x_1, \ldots, x_{\mathsf{r}(g)})]\!]_\tau, e_{u,v}) \leq \sum_{m = \bigoplus_{i=1}^{\mathsf{r}(g)} m_r \odot_{x_i} 1_{x_i}} p_r(m_r) \cdot \lambda \cdot \left(1 - \prod_{i=1}^{\mathsf{r}(g)} \left(1 - \frac{d(u_i, v_i)}{\lambda} \right)^{m_r(x_i)} \right) \tag{5.19}$$

with $p = \bigoplus_{i=1}^{\mathsf{r}(g)} (p_r \odot_{x_i} 1_{x_i})$ for some $r \in R_g$ and some $p_r \in \rho(r)$, by

$$\begin{aligned}
&\mathbf{A}([\![g(x_1, \ldots, x_{\mathsf{r}(g)})]\!]_\tau, e_{u,v}) \\
&= \mathbf{P}(p, e_{u,v}) \\
&= \sum_{m \in \mathscr{M}} p(m) \cdot \mathbf{D}(m, e_{u,v}) \\
&= \sum_{m = \bigoplus_{i=1}^{\mathsf{r}(g)} m_r \odot_{x_i} 1_{x_i}} p_r(m_r) \cdot \mathbf{D}(m_r, e_{u,v})
\end{aligned}$$

$$= \sum_{\substack{m\in\mathscr{M} \\ m=\bigoplus_{i=1}^{\mathsf{r}(g)} m_r \otimes_{x_i} 1_{x_i}}} p_r(m_r)\cdot\lambda\cdot\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{d(u_i,v_i)}{\lambda}\right)^{m_r(x_i)}\right)$$

whereas if $d(u_i,v_i)=1$ for some $1\le i\le \mathsf{r}(g)$, then we have

$$\mathbf{A}([\![g(x_1,\ldots,x_{\mathsf{r}(g)})]\!]_\tau, e_{u,v}) \le 1 \tag{5.20}$$

by

$$\begin{aligned}
&\mathbf{A}([\![g(x_1,\ldots,x_{\mathsf{r}(g)})]\!]_\tau, e_{u,v}) \\
=&\mathbf{P}(p, e_{u,v}) \\
=&\sum_{m\in\mathscr{M}} p(m)\cdot\mathbf{D}(m, e_{u,v}) \\
=&\sum_{\substack{m\in\mathscr{M} \\ m=\bigoplus_{i=1}^{\mathsf{r}(g)} m_r \otimes_{x_i} 1_{x_i}}} p_r(m_r)\cdot\mathbf{D}(m_r, e_{u,v}) \\
=&\sum_{\substack{m\in\mathscr{M} \\ m=\bigoplus_{i=1}^{\mathsf{r}(g)} m_r \otimes_{x_i} 1_{x_i}}} p_r(m_r)\cdot 1 \\
=&1.
\end{aligned}$$

Therefore, the upper bound for the distance between distributions $\sigma_1(\theta)$ and $\sigma_2(\theta)$ given in (the right side of) Equation 5.18 satisfies:

$$\begin{aligned}
&\lambda\cdot\sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)}) \\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})}\in\mathsf{T}(\Sigma)} \omega(u,v)\cdot d(u,v) \\
=&\lambda\cdot\sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)}) \\ v=g(v_1,\ldots,v_{\mathbf{r}(g)}) \\ \forall 1\le i\le\mathbf{r}(g).d(u_i,v_i)<1}\in\mathsf{T}(\Sigma)} \omega(u,v)\cdot d(u,v) + \lambda\cdot\sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)}) \\ v=g(v_1,\ldots,v_{\mathbf{r}(g)}) \\ \exists 1\le i\le\mathbf{r}(g).d(u_i,v_i)=1}\in\mathsf{T}(\Sigma)} \omega(u,v)\cdot d(u,v) \\
\le&\lambda\cdot\sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)}) \\ v=g(v_1,\ldots,v_{\mathbf{r}(g)}) \\ \forall 1\le i\le\mathbf{r}(g).d(u_i,v_i)<1}\in\mathsf{T}(\Sigma)} \omega(u,v)\cdot\sum_{\substack{m\in\mathscr{M} \\ m=\bigoplus_{i=1}^{\mathsf{r}(g)} m_r \otimes_{x_i} 1_{x_i}}} p^{u,v}(m_r)\cdot\lambda\cdot\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{d(u_i,v_i)}{\lambda}\right)^{m_r(x_i)}\right) \\
&\quad+\lambda\cdot\sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)}) \\ v=g(v_1,\ldots,v_{\mathbf{r}(g)}) \\ \exists 1\le i\le\mathbf{r}(g).d(u_i,v_i)=1}\in\mathsf{T}(\Sigma)} \omega(u,v)\cdot 1 \\
\le&\lambda\cdot\sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)}) \\ v=g(v_1,\ldots,v_{\mathbf{r}(g)}) \\ \forall 1\le i\le\mathbf{r}(g).d(u_i,v_i)<1}\in\mathsf{T}(\Sigma)} \omega(u,v)\cdot\sum_{\substack{m\in\mathscr{M} \\ m=\bigoplus_{i=1}^{\mathsf{r}(g)} m_r \otimes_{x_i} 1_{x_i}}} p^{u,v}(m_r)\cdot\lambda\cdot\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{d(u_i,v_i)}{\lambda}\right)^{(m_r\otimes 1_{X_g})(x_i)}\right) \\
&\quad+\lambda\cdot\sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)}) \\ v=g(v_1,\ldots,v_{\mathbf{r}(g)}) \\ \exists 1\le i\le\mathbf{r}(g).d(u_i,v_i)=1}\in\mathsf{T}(\Sigma)} \omega(u,v)
\end{aligned}$$

$$
\begin{aligned}
=&\lambda\cdot \sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})\\ \forall 1\le i\le \mathbf{r}(g).d(u_i,v_i)<1}} \in \mathsf{T}(\Sigma)\ \omega(u,v)\cdot \mathbf{P}(p^{u,v}\otimes 1_{X_g}, e_{u,v}) + \lambda\cdot \sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})\\ \exists 1\le i\le \mathbf{r}(g).d(u_i,v_i)=1}} \in \mathsf{T}(\Sigma)\ \omega(u,v)\\
\le&\lambda\cdot \sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})\\ \forall 1\le i\le \mathbf{r}(g).d(u_i,v_i)<1}} \in \mathsf{T}(\Sigma)\ \omega(u,v)\cdot \mathbf{P}((\otimes_{\omega(u,v)>0} p^{u,v})\otimes 1_{X_g}, e_{u,v}) + \lambda\cdot \sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})\\ \exists 1\le i\le \mathbf{r}(g).d(u_i,v_i)=1}} \in \mathsf{T}(\Sigma)\ \omega(u,v)\\
\le&\lambda\cdot \sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})\\ \forall 1\le i\le \mathbf{r}(g).d(u_i,v_i)<1}} \in \mathsf{T}(\Sigma)\ \omega(u,v)\cdot \mathbf{P}((\otimes_{p\in\bigcup_{r\in R_g}\rho(r)} p)\otimes 1_{X_g}, e_{u,v}) + \lambda\cdot \sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})\\ \exists 1\le i\le \mathbf{r}(g).d(u_i,v_i)=1}} \in \mathsf{T}(\Sigma)\ \omega(u,v)\\
=&\lambda\cdot \sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})\\ \forall 1\le i\le \mathbf{r}(g).d(u_i,v_i)<1}} \in \mathsf{T}(\Sigma)\ \omega(u,v)\cdot \mathbf{P}(\otimes_{r\in R_g}((\otimes_{p\in\rho(r)} p)\otimes 1_{X_r}), e_{u,v})\\
&+\lambda\cdot \sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})\\ \exists 1\le i\le \mathbf{r}(g).d(u_i,v_i)=1}} \in \mathsf{T}(\Sigma)\ \omega(u,v)\\
=&\lambda\cdot \sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})\\ \forall 1\le i\le \mathbf{r}(g).d(u_i,v_i)<1}} \in \mathsf{T}(\Sigma)\ \prod_{i=1}^{\mathbf{r}(g)}\omega_i(u_i,v_i)\cdot \mathbf{P}(\otimes_{r\in R_g}((\otimes_{p\in\rho(r)} p)\otimes 1_{X_r}), e_{u,v})\\
&+\lambda\cdot \sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})\\ \exists 1\le i\le \mathbf{r}(g).d(u_i,v_i)=1}} \in \mathsf{T}(\Sigma)\ \omega(u,v)\\
=&\lambda\cdot \sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})\\ \forall 1\le i\le \mathbf{r}(g).d(u_i,v_i)<1}} \in \mathsf{T}(\Sigma)\ \prod_{i=1}^{\mathbf{r}(g)}\omega_i(u_i,v_i)\cdot \sum_{m\in\mathscr{M}}(\otimes_{r\in R_g}((\otimes_{p\in\rho(r)} p)\otimes 1_{X_r}))(m)\cdot\lambda\cdot\\
&\qquad\qquad\left(1-\prod_{i=1}^{\mathbf{r}(g)}\left(1-\frac{d(u_i,v_i)}{\lambda}\right)^{m(x_i)}\right)\\
&+\quad \lambda\cdot \sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})\\ \exists 1\le i\le \mathbf{r}(g).d(u_i,v_i)=1}} \in \mathsf{T}(\Sigma)\ \omega(u,v)\\
=&\lambda\cdot \sum_{m\in\mathscr{M}}(\otimes_{r\in R_g}((\otimes_{p\in\rho(r)} p)\otimes 1_{X_r}))(m)\cdot \sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})\\ \forall 1\le i\le \mathbf{r}(g).d(u_i,v_i)<1}} \in \mathsf{T}(\Sigma)\ \prod_{i=1}^{\mathbf{r}(g)}\omega_i(u_i,v_i)\cdot\lambda\cdot\\
&\qquad\qquad\left(1-\prod_{i=1}^{\mathbf{r}(g)}\left(1-\frac{d(u_i,v_i)}{\lambda}\right)^{m(x_i)}\right)\\
&+\lambda\cdot \sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})\\ \exists 1\le i\le \mathbf{r}(g).d(u_i,v_i)=1}} \in \mathsf{T}(\Sigma)\ \omega(u,v)
\end{aligned}
$$

$$\leq \sum_{m\in\mathscr{M}} (\mathbb{V}_{r\in R_g}((\mathbb{V}_{p\in\rho(r)}p)\mathbb{V}\,1_{X_r}))(m)\cdot \sum_{\substack{u=g(u_1,\dots,u_{\mathbf{r}(g)})\\ v=g(v_1,\dots,v_{\mathbf{r}(g)})\\ \forall 1\leq i\leq \mathbf{r}(g).d(u_i,v_i)<1}} \in \mathsf{T}(\Sigma) \prod_{i=1}^{\mathsf{r}(g)} \omega_i(u_i,v_i)\cdot\lambda^2\cdot$$

$$\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{d(u_i,v_i)}{\lambda}\right)^{m(x_i)}\right)$$

$$+\lambda^2\cdot \sum_{\substack{u=g(u_1,\dots,u_{\mathbf{r}(g)})\\ v=g(v_1,\dots,v_{\mathbf{r}(g)})\\ \exists 1\leq i\leq \mathbf{r}(g).d(u_i,v_i)=1}} \in \mathsf{T}(\Sigma) \omega(u,v)$$

$$= \sum_{m\in\mathscr{M}} (\mathbb{V}_{r\in R_g}((\mathbb{V}_{p\in\rho(r)}p)\mathbb{V}\,1_{X_r}))(m)\cdot\lambda^2\cdot$$

$$\left(1-\sum_{\substack{u=g(u_1,\dots,u_{\mathbf{r}(g)})\\ v=g(v_1,\dots,v_{\mathbf{r}(g)})\\ \forall 1\leq i\leq \mathbf{r}(g).d(u_i,v_i)<1}} \in \mathsf{T}(\Sigma) \prod_{i=1}^{\mathsf{r}(g)} \omega_i(u_i,v_i)\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{d(u_i,v_i)}{\lambda}\right)^{m(x_i)}\right)$$

$$= \sum_{m\in\mathscr{M}} (\mathbb{V}_{r\in R_g}((\mathbb{V}_{p\in\rho(r)}p)\mathbb{V}\,1_{X_r}))(m)\cdot\lambda^2\cdot$$

$$\left(1-\prod_{i=1}^{\mathsf{r}(g)}\sum_{\substack{u_i,v_i\in\mathbf{T}(\Sigma)\\ d(u_i,v_i)<1}} \omega_i(u_i,v_i)\cdot\left(1-\frac{d(u_i,v_i)}{\lambda}\right)^{m(x_i)}\right)$$

$$\leq \sum_{m\in\mathscr{M}} (\mathbb{V}_{r\in R_g}((\mathbb{V}_{p\in\rho(r)}p)\mathbb{V}\,1_{X_r}))(m)\cdot\lambda^2\cdot$$

$$\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{\sum_{\substack{u_i,v_i\in\mathbf{T}(\Sigma)\\ d(u_i,v_i)=1}}\omega_i(u_i,v_i)\cdot d(u_i,v_i)}{\lambda}\right)^{m(x_i)}\right)$$

$$\leq \sum_{m\in\mathscr{M}} (\mathbb{V}_{r\in R_g}((\mathbb{V}_{p\in\rho(r)}p)\mathbb{V}\,1_{X_r}))(m)\cdot\lambda^2\cdot$$

$$\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{\mathbf{K}(d)(\sigma_1(\theta_i),\sigma_2(\theta_i))-\sum_{\substack{u_i,v_i\in\mathbf{T}(\Sigma)\\ d(u_i,v_i)=1}}\omega_i(u_i,v_i)}{\lambda}\right)^{m(x_i)}\right)$$

$$\leq \sum_{m\in\mathscr{M}} (\mathbb{V}_{r\in R_g}((\mathbb{V}_{p\in\rho(r)}p)\mathbb{V}\,1_{X_r}))(m)\cdot\lambda^2\cdot$$

$$\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{\mathbf{K}(d)(\sigma_1(\theta_i),\sigma_2(\theta_i))}{\lambda}\right)^{m(x_i)}\right)$$

$$\leq \sum_{m\in\mathscr{M}} (\mathbb{V}_{r\in R_g}((\mathbb{V}_{p\in\rho(r)}p)\mathbb{V}\,1_{X_r}))(m)\cdot\lambda^2\cdot$$

$$\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{\mathbf{P}(\hat{p}_i,e')}{\lambda^2}\right)^{m(x_i)}\right)$$

$$= \sum_{m\in\mathscr{M}} (\mathbin{\mathbb{O}}_{r\in R_g}((\mathbin{\mathbb{O}}_{p\in\rho(r)}p)\mathbin{\mathbb{O}} 1_{X_r}))(m)\cdot\lambda^2\cdot$$
$$\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{\sum_{m_i\in\mathscr{M}}\hat{p}_i(m_i)\cdot\mathbf{D}(m_i,e')}{\lambda^2}\right)^{m(x_i)}\right)$$

$$\leq \sum_{m\in\mathscr{M}} (\mathbin{\mathbb{O}}_{r_g\in R_g}((\mathbin{\mathbb{O}}_{p\in\rho(r_g)}p)\mathbin{\mathbb{O}} 1_{X_r}))(m)\cdot\lambda^2\cdot$$
$$\left(1-\prod_{i=1}^{\mathsf{r}(g)}\sum_{m_i\in\mathscr{M}}\hat{p}_i(m_i)\cdot\left(1-\frac{\mathbf{D}(m_i,e')}{\lambda^2}\right)^{m(x_i)}\right)$$

$$\leq \sum_{m\in\mathscr{M}} (\mathbin{\mathbb{O}}_{r_g\in R_g}((\mathbin{\mathbb{O}}_{p\in\rho(r_g)}p)\mathbin{\mathbb{O}} 1_{X_r}))(m)\cdot\lambda\cdot$$
$$\left(1-\prod_{i=1}^{\mathsf{r}(g)}\sum_{m_i\in\mathscr{M}}\hat{p}_i(m_i)\cdot\left(1-\frac{\mathbf{D}(m_i,e')}{\lambda}\right)^{m(x_i)}\right)$$

where step 1 is immediate, step 2 follows from Equations 5.19 and 5.20, step 3 follows from inequality $(1-(d(u_i,v_i)/\lambda))<1$, step 4 follows immediately by the definition of $\mathbf{P}$, both steps 5 and 6 follow by monotonicity of $\mathbf{P}$ (Lemma 5.19), step 7 follows by associativity of $\mathbin{\mathbb{O}}$ (Lemma 5.29), step 8 follows immediately by $\omega(u,v)=\prod_{i=1}^{\mathsf{r}(g)}\omega_i(u_i,v_i)$, step 9 follows immediately by the definition of $\mathbf{P}$, both steps 10 and 11 are immediate, step 12 follows immediately by $\omega(u,v)=\prod_{i=1}^{\mathsf{r}(g)}\omega_i(u_i,v_i)$, step 13 is easily provable by induction over $\mathsf{r}(g)$, step 14 follows by Jensen's inequality (which can be applied since the function $h(x)=1-(x/\lambda)^{m(x)}$ with $m(x)\geq 1$ or $m(x)=0$ is concave in $[0,\lambda]$, and all multiplicities m in the support of $\mathbin{\mathbb{O}}_{r\in R_g}((\mathbin{\mathbb{O}}_{p\in\rho(r)}p)\mathbin{\mathbb{O}} 1_{X_r})$ are such that $m(x_i)\geq 1$ or $m(x_i)=0$ because of factor 1_{X_r}), step 15 follows immediately by the definition of $\mathbf{K}$, step 16 is immediate, step 17 follows by the inductive hypothesis (Equation 5.17), step 18 follows immediately by the definition of $\mathbf{P}$, step 19 follows by Jensen's inequality (which can be applied since $h(x)=1-(x/\lambda)^{m_i(x)}$ with $m_i(x)>1$ is concave, $\mathbf{D}(m_i,e')$ is concave and, therefore, their composition is convex)., step 20 is immediate. Summarizing:

$$\lambda\cdot\sum_{\substack{u=g(u_1,\ldots,u_{\mathbf{r}(g)})\\ v=g(v_1,\ldots,v_{\mathbf{r}(g)})}\in\mathsf{T}(\Sigma)}\omega(u,v)\cdot d(u,v)\leq$$
$$\sum_{m\in\mathscr{M}} (\mathbin{\mathbb{O}}_{r_g\in R_g}((\mathbin{\mathbb{O}}_{p\in\rho(r_g)}p)\mathbin{\mathbb{O}} 1_{X_r}))(m)\cdot\lambda\cdot\left(1-\prod_{i=1}^{\mathsf{r}(g)}\sum_{m_i\in\mathscr{M}}\hat{p}_i(m_i)\cdot\left(1-\frac{\mathbf{D}(m_i,e')}{\lambda}\right)^{m(x_i)}\right) \tag{5.21}$$

Now we consider the right hand side $\mathbf{P}(\hat{p},e')$ of the proof obligation in Equation 5.17. By Lemma 5.29 and equations 5.13–5.15 the unique $\hat{p}\in\mathrm{src}(\tau(\theta),r)$ can be reformulated to:

$$
\begin{aligned}
\hat{p} &= \mathrm{src}\left(\bigoplus_{i=1}^{\mathsf{r}(g)}\left(\left(\mathbin{\mathbb{O}}_{r_g\in R_g}\left(\left(\mathbin{\mathbb{O}}_{p\in\rho(r_g)}p\right)\mathbin{\mathbb{O}}1_{X_{r_g}}\right)\right)\odot_{x_i}p_i\right),r\right)\\
&=\bigoplus_{i=1}^{\mathsf{r}(g)}\mathrm{src}\left(\left(\left(\mathbin{\mathbb{O}}_{r_g\in R_g}\left(\left(\mathbin{\mathbb{O}}_{p\in\rho(r_g)}p\right)\mathbin{\mathbb{O}}1_{X_{r_g}}\right)\right)\odot_{x_i}p_i\right),r\right)\\
&=\bigoplus_{i=1}^{\mathsf{r}(g)}\left(\left(\mathbin{\mathbb{O}}_{r_g\in R_g}\left(\left(\mathbin{\mathbb{O}}_{p\in\rho(r_g)}p\right)\mathbin{\mathbb{O}}1_{X_{r_g}}\right)\right)\odot_{x_i}\mathrm{src}(p_i,r)\right)\\
&=\bigoplus_{i=1}^{\mathsf{r}(g)}\left(\left(\mathbin{\mathbb{O}}_{r_g\in R_g}\left(\left(\mathbin{\mathbb{O}}_{p\in\rho(r_g)}p\right)\mathbin{\mathbb{O}}1_{X_{r_g}}\right)\right)\odot_{x_i}\hat{p}_i\right)
\end{aligned}
$$

with $p_i\in\tau(\theta_i)$ such that $\hat{p}_i=\mathrm{src}(p_i,r)$ for $i=1,\dots,\mathsf{r}(g)$. Hence by exploiting the results on distributions of operators over **D** and **P** in Lemma 5.30 and Lemma 5.31 we get

$$
\begin{aligned}
&\mathbf{P}(\hat{p},e')\\
&=\mathbf{P}\left(\bigoplus_{i=1}^{\mathsf{r}(g)}\left(\left(\mathbin{\mathbb{O}}_{r_g\in R_g}\left(\left(\mathbin{\mathbb{O}}_{p\in\rho(r_g)}p\right)\mathbin{\mathbb{O}}1_{X_r}\right)\right)\odot_{x_i}\hat{p}_i\right),e'\right)\\
&=\sum_{\substack{m_i\in\mathscr{M}\\ i=1,\dots\mathbf{r}(g)}}\prod_{i=1}^{\mathsf{r}(g)}\left(\left(\mathbin{\mathbb{O}}_{r_g\in R_g}\left(\left(\mathbin{\mathbb{O}}_{p\in\rho(r_g)}p\right)\mathbin{\mathbb{O}}1_{X_r}\right)\right)\odot_{x_i}\hat{p}_i\right)(m_i)\cdot\lambda\cdot\\
&\qquad\qquad\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{\mathbf{D}(m_i,e')}{\lambda}\right)\right)\\
&=\sum_{\substack{m_i',m_i''\in\mathscr{M}\\ i=1,\dots\mathbf{r}(g)}}\prod_{i=1}^{\mathsf{r}(g)}\left(\mathbin{\mathbb{O}}_{r_g\in R_g}\left(\left(\mathbin{\mathbb{O}}_{p\in\rho(r_g)}p\right)\mathbin{\mathbb{O}}1_{X_r}\right)(m_i')\cdot\hat{p}_i(m_i'')\right)\cdot\lambda\cdot\\
&\qquad\qquad\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{\mathbf{D}(m_i'\odot_{x_i}m_i'',e')}{\lambda}\right)\right)\\
&\geq\sum_{\substack{m',m_i''\in\mathscr{M}\\ i=1,\dots\mathbf{r}(g)}}\prod_{i=1}^{\mathsf{r}(g)}\left(\mathbin{\mathbb{O}}_{r_g\in R_g}\left(\left(\mathbin{\mathbb{O}}_{p\in\rho(r_g)}p\right)\mathbin{\mathbb{O}}1_{X_r}\right)(m')\cdot\hat{p}_i(m_i'')\right)\cdot\lambda\cdot\\
&\qquad\qquad\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{\mathbf{D}(m'\odot_{x_i}m_i'',e')}{\lambda}\right)\right)\\
&=\sum_{\substack{m',m_i''\in\mathscr{M}\\ i=1,\dots\mathbf{r}(g)}}\prod_{i=1}^{\mathsf{r}(g)}\left(\mathbin{\mathbb{O}}_{r_g\in R_g}\left(\left(\mathbin{\mathbb{O}}_{p\in\rho(r_g)}p\right)\mathbin{\mathbb{O}}1_{X_r}\right)(m')\cdot\hat{p}_i(m_i'')\right)\cdot\lambda\cdot
\end{aligned}
$$

$$\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{\lambda\cdot\left(1-\left(1-\frac{\mathbf{D}(m_i'',e')}{\lambda}\right)^{m'(x_i)}\right)}{\lambda}\right)\right)$$

$$=\sum_{\substack{m',m_i''\in\mathscr{M}\\ i=1,\dots\mathbf{r}(g)}}\prod_{i=1}^{\mathsf{r}(g)}\left(\mathbb{O}_{r_g\in R_g}\left(\left(\mathbb{O}_{p\in\rho(r_g)}p\right)\mathbb{O}\,1_{X_r}\right)\right)(m')\cdot\hat{p}_i(m_i'')\cdot\lambda\cdot$$

$$\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{\mathbf{D}(m_i'',e')}{\lambda}\right)^{m'(x_i)}\right)$$

$$=\sum_{m'\in\mathscr{M}}\left(\mathbb{O}_{r_g\in R_g}\left(\left(\mathbb{O}_{p\in\rho(r_g)}p\right)\mathbb{O}\,1_{X_r}\right)\right)(m')\cdot\sum_{\substack{m_i''\in\mathscr{M}\\ i=1,\dots\mathbf{r}(g)}}\prod_{i=1}^{\mathsf{r}(g)}\hat{p}_i(m_i'')\cdot\lambda\cdot$$

$$\left(1-\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{\mathbf{D}(m_i'',e')}{\lambda}\right)^{m'(x_i)}\right)$$

$$=\sum_{m'\in\mathscr{M}}\left(\mathbb{O}_{r_g\in R_g}\left(\left(\mathbb{O}_{p\in\rho(r_g)}p\right)\mathbb{O}\,1_{X_r}\right)\right)(m')\cdot\lambda\cdot$$

$$\left(1-\sum_{\substack{m_i''\in\mathscr{M}\\ i=1,\dots\mathbf{r}(g)}}\prod_{i=1}^{\mathsf{r}(g)}\hat{p}_i(m_i'')\cdot\prod_{i=1}^{\mathsf{r}(g)}\left(1-\frac{\mathbf{D}(m_i'',e')}{\lambda}\right)^{m'(x_i)}\right)$$

$$=\sum_{m'\in\mathscr{M}}\left(\mathbb{O}_{r_g\in R_g}\left(\left(\mathbb{O}_{p\in\rho(r_g)}p\right)\mathbb{O}\,1_{X_r}\right)\right)(m')\cdot\lambda\cdot$$

$$\left(1-\prod_{i=1}^{\mathsf{r}(g)}\sum_{m_i''\in\mathscr{M}}\hat{p}_i(m_i'')\cdot\left(1-\frac{\mathbf{D}(m_i'',e')}{\lambda}\right)^{m'(x_i)}\right)$$

where step 1 follows by equality $\hat{p}=\bigoplus_{i=1}^{\mathsf{r}(g)}\left(\left(\mathbb{O}_{r_g\in R_g}\left(\left(\mathbb{O}_{p\in\rho(r_g)}p\right)\mathbb{O}\,1_{X_{r_g}}\right)\right)\odot_{x_i}\hat{p}_i\right)$, step 2 follows by how $\oplus$ distributes over $\mathbf{P}$ (Lemma 5.31.1), step 3 follows by definition of $\odot_{x_i}$ on $\mathscr{P}$, step 4 is immediate, step 5 follows by how $\odot$ distributes over $\mathbf{D}$ (Lemma 5.30.2), and steps 6–9 are immediate.

Summarizing

$$\mathbf{P}(\hat{p},e')\geq$$

$$\sum_{m'\in\mathscr{M}}\left(\mathbb{O}_{r_g\in R_g}\left(\left(\mathbb{O}_{p\in\rho(r_g)}p\right)\mathbb{O}\,1_{X_r}\right)\right)(m')\cdot\lambda\cdot\left(1-\prod_{i=1}^{\mathsf{r}(g)}\sum_{m_i''\in\mathscr{M}}\hat{p}_i(m_i'')\cdot\left(1-\frac{\mathbf{D}(m_i'',e')}{\lambda}\right)^{m'(x_i)}\right)\tag{5.22}$$

We conclude by observing that the proof obligation Equation 5.17 follows from Equations 5.18, 5.21, 5.22. □

Inductive consistency of **F** and **B** (Proposition 5.44) and monotonicity and continuity of **F** and **B** (Proposition 5.39) allows to conclude now with the following adequacy

theorem.

Theorem 5.45 (Adequacy of term denotations)**.** *Let P be any PGSOS PTSS with* $\mathbf{d}$ *the bisimilarity metric on the associated PTS and* $[\![\cdot]\!]$ *the canonical denotation of terms according to P. Then* $\mathbf{d} \preceq [\![\cdot]\!]$.

Proof. Recall that $\mathbf{d}$ is the least fixed point of $\mathbf{B}\colon [0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)} \to [0,1]^{\mathsf{T}(\Sigma)\times\mathsf{T}(\Sigma)}$ defined by $\mathbf{B}(d)(t,t') = \sup_{a\in A}\{\lambda \cdot \mathbf{H}(\mathbf{K}(d))(der(t,a), der(t',a))\}$. Let $d_n = \mathbf{B}^n(\mathbf{0})$ and $(\tau_n, \rho_n) = \mathbf{F}^n(\bot_T, \bot_R)$ for all $n \in \mathbb{N}$. Note that $\mathbf{d} = \lim_{n\to\infty} d_n$ and $(\omega_T, \omega_R) = \lim_{n\to\infty}(\tau_n, \rho_n)$.

First, we show that $d_n \preceq [\![\cdot]\!]_{\tau_n}$ for all $n \in \mathbb{N}$. We proceed by induction over n. For $n = 0$ this is trivial because $d_0 = \mathbf{0}$. For any finite n it follows by induction from Proposition 5.44 together with monotonicity of $\mathbf{B}$ [DD11] and monotonicity of $\mathbf{F}$ (Proposition 5.39).

Then, monotonicity and upward ω-continuity of $\mathbf{F}$ (Proposition 5.39) ensure that this property is also preserved in the limit, namely $\mathbf{d} \preceq [\![\cdot]\!]$. In detail, let $t \in \mathbb{T}(\Sigma)$ be any term and σ_1, σ_2 any closed substitutions. Consider the sequence over reals $[d_n(\sigma_1(t), \sigma_2(t))]_{n\in\mathbb{N}}$. By monotonicity of $\mathbf{B}$ [DD11] this sequence is a monotone increasing chain and by Proposition 2.25 it has $\mathbf{d}(\sigma_1(t), \sigma_2(t))$ as supremum. By $d_n \sqsubseteq [\![\cdot]\!]_{\tau_n}$ we have $d_n(\sigma_1(t), \sigma_2(t)) \leq \mathbf{A}([\![t]\!]_{\tau_n}, \mathbf{d}(\sigma_1, \sigma_2))$. From $\tau_n \sqsubseteq \omega_T$ and monotonicity of $\mathbf{A}$ (Proposition 5.26) it follows $d_n(\sigma_1(t), \sigma_2(t)) \leq \mathbf{A}([\![t]\!], \mathbf{d}(\sigma_1, \sigma_2))$. Summarizing, we have that $\mathbf{A}([\![t]\!], \mathbf{d}(\sigma_1, \sigma_2))$ is an upper bound to all elements in the chain $[d_n(\sigma_1(t), \sigma_2(t))]_{n\in\mathbb{N}}$, whose supremum is $\mathbf{d}(\sigma_1(t), \sigma_2(t))$. Then $\mathbf{d}(\sigma_1(t), \sigma_2(t)) \leq \mathbf{A}([\![t]\!], \mathbf{d}(\sigma_1, \sigma_2))$ follows immediately. Hence, we conclude $\mathbf{d} \preceq [\![\cdot]\!]$. □

5.3.4 Discussion

We conclude this section by discussing the connection between the operational behavior of processes and the operations and functional expressions to compute the process denotations. Table 5.1 shows how basic process behavior and process combinators relate to basic denotations and operations on denotations. The functional $\mathbf{F}$ to compute the denotation of some term t applies then compositionally the operations on denotations to subterms of t mimicking the compositional term structure of t.

The distance between replicated processes depends on the interaction between the process instances. Nondeterministic choice between two instances of a process with deterministic initial state $x + x$ can be understood as independence between the two instances of x which results in $[\![x + x]\!] = [\![x]\!] = 1_x$. However, nondeterministic choices between two instances of a process with probabilistic initial state $[\![\mu + \mu]\!] = (\rho_+ \odot_{x_1} \mu) \oplus (\rho_+ \odot_{x_2} \mu) = 2_\mu$ (cf. Example 5.38). On the other hand, if processes copies (with deterministic or probabilistic initial state) synchronize, then $[\![x \parallel x]\!] = 2[\![x]\!] = 2_x$ and $[\![\mu \parallel \mu]\!] = (\rho_\parallel \odot_{x_1} \mu) \oplus (\rho_\parallel \odot_{x_2} \mu) = 2_\mu$. Since the denotational model stratifies deterministic behavior, probabilistic choice and nondeterministic choice, the process algebra operators for probabilistic and nondeterministic choice are expressed directly as operations on the probabilistic multiplicity (probabilistic choice) and nondeterministic probabilistic multiplicity (nondeterministic choice). Sequencing and evolution of processes are represented by scalar multiplication with the discount factor since these operations alter after how many steps the behavioral difference is observable.

The denotations for process creation, process replication, probabilistic choice and nondeterministic choice form a basis of the set of all finite denotations.

Process behavior	BPA expression	Operation on denotation
Process creation	x	1_x
Process evolution	$a.t$	$\lambda \cdot [\![t]\!]$
Process sequencing	$t_1; t_2$	$[\![t_1]\!] \oplus \lambda \cdot (\bigoplus_{x \in \mathcal{V}_s} ([\![t_2]\!] \odot_x 1_x)) \cup [\![t_2]\!]$
Process replication	$t \parallel t$	$[\![t]\!] \oplus [\![t]\!]$
Nondeterministic process choice	$t_1 + t_2$	$[\![t_1]\!] \cup [\![t_2]\!]$
Probabilistic process choice	$a.([p_1]t_1 \oplus [p_2]t2)$	$\lambda \cdot p_1 [\![t_1]\!] + \lambda \cdot p_2 [\![t_2]\!]$

Table 5.1: Compositional process term structure and compositional process denotations

Theorem 5.46. *Consider the constant denotations* $\{0_x, 1_x | x \in \mathcal{V}_s\}$. *The inductive application of the unary operations* $\{\lambda \cdot \_ | \lambda \in (0,1]\}$ *and the binary operations* $\{\_ \oplus \_, \_ \cup \_, \_ + \_\}$ *form the set of all finite denotations* $\mathcal{D}_{fin} = \{\downarrow P | P \subseteq \mathcal{P} \wedge \forall p \in P. \overline{p}(x) < \infty\}$.

Proof. Consider first an arbitrary multiplicity $m \in \mathcal{M}$ with $m(x) < \infty$ for all $x \in \mathcal{V}_s$. We have

$$m = \left(\bigoplus_{\{x | m(x) = 0\}} 0_x \right) \oplus \left(\bigoplus_{\{x | m(x) > 0\}} m(x) \cdot 1_x \right)$$

which confirms that we can build m by using constants 0_x and 1_x and the operators $\cdot$ and $\oplus$. Consider now an arbitrary probabilistic multiplicity $p \in \mathcal{P}$ with $\overline{p}(x) < \infty$. From $\overline{p}(x) < \infty$ we infer $m(x) < \infty$ for all $x \in \mathcal{V}_s$ and $m \in \mathcal{M}$ with $p(m) > 0$, thus implying that any m in the support of p can be built by 0_x and 1_x and the operators $\cdot$ and $\oplus$. Hence $p = \sum_{\{m \in \mathcal{M} | p(m) > 0\}} p(m) \cdot m$, thus implying that p can be built by using 0_x, 1_x and opertors $\cdot$, $\oplus$ and $+$. Finally, by using operator $\cup$ we can build all nondeterministic probabilistic multiplicities in $\mathcal{D}_{\text{fin}}$. □

We conclude by remarking that the denotational model of PTSSs (Definition 5.34) could be simplified (with the necessary adaptation of the functional **F**) to $(S_T, \sqsubseteq)$ since $\tau_k(f(x_1, \ldots, x_n)) = \bigcup_{r \in R_f} \rho_k(r)$ for all $k \in \mathbb{N}$. However, we prefer the presented clear separation between the denotation of terms S_T and the denotation of the rules defining the operational semantics of operators S_R.

5.4 Compositional reasoning

Uniformly continuous operators allow to reason in a compositional manner (cf. Chapter 3). We explore now in Section 5.4.1 how to specify uniformly continuous operators of a given modulus of continuity and how to derive from the specification rules of an operator its modulus of continuity by using the denotation of the respective operators. Then we study in Section 5.4.2 the composition of those results and derive from the denotation of any term (syntactic composition of operators) its modulus of continuity by functional composition of the moduli of continuity of the operators.

5.4.1 Compositional operators

Uniformly continuous operators are operators that admit a modulus of continuity (Definitions 4.30 and 4.31).

The distance approximation **A** applied to the denotation $[\![f(x_1,\ldots,x_{\mathsf{r}(f)})]\!]$ of the term $f(x_1,\ldots,x_{\mathsf{r}(f)})$ gives an upper bound on the distance between f-composed terms. Moreover, this upper bound is a candidate for the modulus of continuity of f.

Definition 5.47 (Denotation induced upper bound)**.** For an operator $f \in \Sigma$, the *upper bound induced by the canonical denotation of* f is the mapping $z_f : [0,1]^{\mathsf{r}(f)} \to [0,1]$ defined by

$$z_f(\epsilon_1,\ldots,\epsilon_{\mathsf{r}(f)}) = \mathbf{A}([\![f(x_1,\ldots,x_{\mathsf{r}(f)})]\!], e)$$

with $e \in \mathscr{E}$ defined by $e(x_i) = \epsilon_i$ for $i = 1,\ldots,\mathsf{r}(f)$ and $e(y) = 0$ otherwise.

Definition 5.41 and Theorem 5.45 gives $\mathbf{d}(\sigma_1(f(x_1,\ldots,x_{\mathsf{r}(f)}),\sigma_2(f(x_1,\ldots,x_{\mathsf{r}(f)}))) \leq \mathbf{A}([\![f(x_1,\ldots,x_{\mathsf{r}(f)})]\!],\mathbf{d}(\sigma_1,\sigma_2))$ for all closed substitutions σ_1,σ_2. Hence z_f is an upper bound for the distance between f-composed terms. However, in general there is no guarantee that z_f is a modulus of continuity for f, since z_f may not be continuous at $(0,\ldots,0)$. As an example, we show that for the replication operator ! of π-calculus it holds that the upper bound $z_!$ is a modulus of continuity if and only if $\lambda < 1$.

Example 5.48. The replication operator of π-calculus is specified by the rule

$$\frac{x \xrightarrow{a} \mu}{!x \xrightarrow{a} \mu \parallel \delta(!x)}$$

Consider first $\lambda = 1$ (non-discounting bisimulation metric). We have $[\![!x]\!] = \infty_x$, which, intuitively, expresses that the argument x is infinitely often replicated. This denotation leads to the upper bound $z_!(e(x)) = \mathbf{A}(\infty_x, e)$, which is not continuous at 0 since $z_!(e(x)) = 1$ if $e(x) > 0$ and $z_!(e(x)) = 0$ if $e(x) = 0$. In general, no modulus of continuity for operator ! can be provided. From $\mathbf{d}(!s,!t) = 1$ whenever $\mathbf{d}(s,t) > 0$, any mapping $z : [0,1] \to [0,1]$ satisfying $\mathbf{d}(!s,!t) \leq z(\mathbf{d}(s,t))$ for all closed terms $s,t \in \mathsf{T}(\Sigma)$ is such that $z(\epsilon) = 1$ whenever $\epsilon > 0$, which implies that there is no mapping $z : [0,1] \to [0,1]$ with $\mathbf{d}(!s,!t) \leq z(\mathbf{d}(s,t))$ for all closed terms $s,t \in \mathsf{T}(\Sigma)$ and $z(0) = 0$ that is also continuous at 0. Hence, the replication operator is not uniformly continuous w.r.t. non-discounting bisimulation metric semantics.

Consider $\lambda < 1$ (discounting bisimulation metric). Let $(\tau_n,\rho_n) = \mathbf{F}^n(\bot_T,\bot_R)$. Then $\tau_{n+1}(!x) = 1_x \oplus \lambda \cdot \tau_n(!x) = (\sum_{i=0}^{n-1} \lambda^i)_x$. Hence, the least fixed point is $\omega_T(!x) = [\![!x]\!] = \bigoplus_{i=0}^{\infty}(\lambda^i)_x = (1/(1-\lambda))_x$. Thus, $\mathbf{A}([\![!x]\!],e) = z_!(e(x)) = \lambda(1-(1-e(x)/\lambda)^{1/(1-\lambda)})$ is a modulus of continuity for the replication operator w.r.t. discounting bisimulation metric semantics. The optimal modulus of continuity for the replication operator is $z(\epsilon) = \epsilon/(1-(\lambda-\epsilon))$ (Proposition 3.9.d)

On the other hand, probabilistic replication operator $!^q\_$ is continuous even for the non-discounted bisimulation metric, witnessed by the modulus of continuity $z_{!^q\_}$.

Example 5.49. We consider now the probabilistic recursion operator $!^q\_$ with $q \in (0,1) \cap \mathbb{Q}$ defined by the rule [MS13]:

$$\frac{x \xrightarrow{a} \mu}{!^q x \xrightarrow{a} \mu \oplus_q (\mu \parallel \delta(!^q x))}$$

Let $(\tau_n, \rho_n) = \mathbf{F}^n(\bot_T, \bot_R)$. Then $\tau_{n+1}(!^q x) = \{[q]1_x + [1-q](1_x \oplus \lambda \cdot \tau_n(!^q x))\}$. The probabilistic replication operator has the denotation $[\![!^q x]\!] = \{p\}$ with $p((\sum_{i=0}^{n-1} \lambda^i)_x) = q(1-q)^{n-1}$ for all $n \geq 1$. Intuitively, $q(1-q)^{n-1}$ is the likelihood that for $n-1$ transitions the probabilistic choice in the rule target resolves to the right summand $\mu \parallel \delta(!^q x)$ followed by a transition where then the probabilistic choice resolves to the left summand μ. Note that $\sum_{i=0}^{n-1} \lambda^i = (1-\lambda^n)/(1-\lambda)$ if $\lambda < 1$ and $\sum_{i=0}^{n-1} \lambda^i = n$ if $\lambda = 1$.

Consider $\lambda = 1$ (non-discounting bisimulation metric). Now $[\![!^q x]\!] = \{p\}$ with $p(n_x) = q(1-q)^{n-1}$ for all $n \geq 1$. Then, we get $\mathbf{A}([\![!^q x]\!], \mathbf{d}(\sigma_1, \sigma_2)) = \sum_{n=1}^{\infty} q(1-q)^{n-1} \lambda(1 - (1 - \mathbf{d}(\sigma_1(x), \sigma_2(x))/\lambda)^n) \leq \mathbf{d}(\sigma_1(x), \sigma_2(x)) q \sum_{n=1}^{\infty} (1-q)^{n-1} n = \mathbf{d}(\sigma_1(x), \sigma_2(x)) \cdot q \cdot \lim_{m\to\infty}(1-(1-q)^{m+1})/q^2 - ((m+1)(1-q)^m)/q = \mathbf{d}(\sigma_1(x), \sigma_2(x)) \cdot q(1/q^2 - 0) = (1/q) \cdot \mathbf{d}(\sigma_1(x), \sigma_2(x))$. Hence, for $q > 0$ the probabilistic replication operator is uniformly continuous. Theorem 5.55 confirms this also by $[\![!^q x]\!](x) \sqsubseteq r_x$ with $r = \sum_{n=1}^{\infty} n \cdot q(1-q)^{n-1} = 1/q$.

Consider $\lambda < 1$ (discounting bisimulation metric). Let $p_1, p_2 \in \mathscr{P}$ be defined by $p_1(n_x) = q(1-q)^{n-1}$ (probabilistic multiplicity w.r.t. non-discounting bisimulation metric) and $p_2((\sum_{i=0}^{n-1} \lambda^i)_x) = q(1-q)^{n-1}$ (probabilistic multiplicity w.r.t. discounting bisimulation metric) for any $n \geq 1$. Then, we have $p_2 \sqsubseteq p_1$ by the matching $\omega \in \Omega(p_1, p_2)$ defined by $\omega((\sum_{i=0}^{n-1} \lambda^i)_x, n_x) = q(1-q)^{n-1}$. Then, by monotonicity of $\mathbf{A}$ (Proposition 5.26) we get $\mathbf{A}(\{p_2\}, e) \leq \mathbf{A}(\{p_1\}, e)$ for all $e \in \mathscr{E}$. Hence, the probabilistic replication operator is uniformly continuous w.r.t. discounting bisimulation metric. The optimal modulus of continuity for the probabilistic replication operator is $z(\epsilon) = \epsilon/(1-(1-q)(\lambda-\epsilon))$ (Proposition 3.9.g).

We aim to derive a sufficient condition that the denotation induced upper bound z_f is also continuous at $(0, \ldots, 0)$, thus implying that z_f is a modulus of continuity for f and f is a uniformly continuous operator.

We start with analyzing which multiplicities $m \in \mathscr{M}$ induce a deterministic upper bound distance approximation $\mathbf{D}(m, \cdot)$ that is continuous at the process distance $0 \in \mathscr{E}$. To this purpose, we need a notion of distance between process distances. We define the distance $d_{\mathscr{E}} \colon \mathscr{E} \times \mathscr{E} \to [0,1]$ by $d_{\mathscr{E}}(e_1, e_2) = \sup_{x \in \mathscr{V}} |e_1(x) - e_2(x)|$ for all $e_1, e_2 \in \mathscr{E}$.

Definition 5.50 (Finite multiplicity)**.** A multiplicity $m \in \mathscr{M}$ is *finite* iff the set $V_m = \{x \in \mathscr{V} \mid m(x) > 0\}$ is finite and $m(x) < \infty$ for all $x \in V_m$.

The deterministic distance approximation $\mathbf{D}(m, \cdot)$ of finite multiplicities $m \in \mathscr{M}$ is continuous at $0 \in \mathscr{E}$.

Proposition 5.51. *If a multiplicity $m \in \mathscr{M}$ is finite, then the function $z \colon \mathscr{E} \to [0,1]$ defined by $z(e) = \mathbf{D}(m, e)$ is continuous at $0 \in \mathscr{E}$.*

Proof. There exists a natural $l \in \mathbb{N}$ such that $m \sqsubseteq l_{V_m}$ with $V_m = \{x \in \mathscr{V} \mid m(x) > 0\}$. We have to show that for each $\epsilon > 0$ there exists a neighborhood of $0 \in \mathscr{E}$ such that for

all $e \in \mathscr{E}$ in this neighborhood we have $z(e) < \epsilon$. Formally, we have to prove that for each $\epsilon > 0$ there exists a $\delta > 0$ such that for all $e \in \mathscr{E}$, if $d_{\mathscr{E}}(e, 0) < \delta$ then $z(e) < \epsilon$. Assume an arbitrary $\epsilon > 0$. Let $\delta = \epsilon/(l \cdot |V_m|)$. For any $e \in \mathscr{E}$ with $d_{\mathscr{E}}(e, 0) < \delta$, i.e. $\sup_{x \in \mathscr{V}} e(x) < \epsilon/(l \cdot |V_m|)$, we get

$$
\begin{aligned}
& z(e) \\
=& \mathbf{D}(m, e) \\
=& \lambda \left(1 - \prod_{x \in \mathscr{V}} \left(1 - \frac{e(x)}{\lambda} \right)^{m(x)} \right) \\
=& \lambda \left(1 - \prod_{\substack{x \in \mathscr{V} \\ m(x) \geq 1}} \left(1 - \frac{e(x)}{\lambda} \right)^{m(x)} \cdot \prod_{\substack{x \in \mathscr{V} \\ 0 \leq m(x) < 1}} \left(1 - \frac{e(x)}{\lambda} \right)^{m(x)} \right) \\
\leq& \lambda \left(1 - \prod_{\substack{x \in \mathscr{V} \\ m(x) \geq 1}} \left(1 - \frac{e(x)}{\lambda} \right)^{m(x)} \cdot \prod_{\substack{x \in \mathscr{V} \\ 0 \leq m(x) < 1}} \left(1 - \frac{e(x)}{\lambda} \right) \right) \\
\leq& \sum_{\substack{x \in \mathscr{V} \\ m(x) \geq 1}} m(x) \cdot e(x) + \sum_{\substack{x \in \mathscr{V} \\ 0 \leq m(x) < 1}} e(x) \\
\leq& \sum_{x \in \mathscr{V}} l \cdot e(x) \\
\leq& l \cdot |V_m| \cdot \sup_{x \in \mathscr{V}} e(x) \\
<& l \cdot |V_m| \cdot (\epsilon/(l \cdot |V_m|)) \\
=& \epsilon.
\end{aligned}
$$

This confirms that z is continuous at $0 \in \mathscr{E}$. □

Next, we derive from a probabilistic multiplicity $p \in \mathscr{P}$ a multiplicity $\overline{p} \in \mathscr{M}$ that induces a deterministic distance approximation that is above the probabilistic distance approximation induced by p, namely $\mathbf{P}(p, e) \leq \mathbf{D}(\overline{p}, e)$ for all $e \in \mathscr{E}$.

Definition 5.52 (Weighting of a probabilistic multiplicity)**.** The *weighting of a probabilistic multiplicity* $p \in \mathscr{P}$ is the multiplicity $\overline{p} \in \mathscr{M}$ defined by

$$\overline{p}(x) = \sum_{m \in \mathscr{M}} p(m) \cdot m(x)$$

for all $x \in \mathscr{V}$.

Intuitively, the number of process copies $m(x)$ are weighted by the probability $p(m)$ of realization of multiplicity m.

Proposition 5.53. *For all probabilistic multiplicities $p \in \mathscr{P}$ and process distances $e \in \mathscr{E}$*

$$\mathbf{P}(p, e) \leq \mathbf{D}(\overline{p}, e).$$

Proof.

$$\begin{aligned}
&\mathbf{P}(p,e)\\
=&\sum_{m\in\mathscr{M}} p(m)\lambda\left(1-\prod_{x\in\mathscr{V}_s}\left(1-\frac{e(x)}{\lambda}\right)^{m(x)}\right)\\
\leq&\lambda\left(1-\prod_{x\in\mathscr{V}}\left(1-\frac{e(x)}{\lambda}\right)^{\sum_{m\in M} p(m)m(x)}\right) && \text{(Jensen's inequality)}\\
=&\lambda\left(1-\prod_{x\in\mathscr{V}}\left(1-\frac{e(x)}{\lambda}\right)^{\overline{p}}\right)\\
=&\mathbf{D}(\overline{p},e)
\end{aligned}$$

and Jensen's inequality is applicable since the function $m\mapsto\lambda\left(1-\prod_{x\in\mathscr{V}}\left(1-\frac{e(x)}{\lambda}\right)^{m(x)}\right)$ with typing $\mathscr{M}\to[0,1]$ is concave for all $\lambda\in(0,1]$. □

Definition 5.54 (Canonical deterministic denotation of operators)**.** For any operator $f\in\Sigma$ we define the *canonical deterministic denotation of operator* f as

$$m_f=\overline{\sup[\![f(x_1,\ldots,x_{\mathrm{r}(f)})]\!]}$$

We may call the canonical deterministic denotation of f also the derived multiplicity of f.

An operator f is uniformly continuous if the canonical deterministic denotation m_f is finite.

Theorem 5.55 (Uniformly continuous operator)**.** *Let $f\in\Sigma$ be any operator. If the canonical deterministic denotation m_f of f is finite, then f is uniformly continuous.*

Proof. This is a special case of Theorem 5.63 below. In detail, $m_f=m_{f(x_1,\ldots,x_{\mathrm{r}(f)})}$ for $m_{f(x_1,\ldots,x_{\mathrm{r}(f)})}$ is the canonical deterministic multiplicity of term $f(x_1,\ldots,x_{\mathrm{r}(f)})$ (Definition 5.62). Theorem 5.63 shows that if $m_{f(x_1,\ldots,x_{\mathrm{r}(f)})}$ is finite then context $f(x_1,\ldots,x_{\mathrm{r}(f)})$ is uniformly continuos (Definition 5.58) and Proposition 5.59 shows that operator f is uniformly continuous iff context $f(x_1,\ldots,x_{\mathrm{r}(f)})$ is uniformly continuos. □

In reverse, for a given modulus of continuity z (understood as the semantical specification of some operator f), we can derive the maximal replication of f-composed processes (understood as the syntactical specification) s.t. any operational specification of f that has a denotation below that maximal replication admits z as modulus of continuity.

Definition 5.56 (Distance bound induced multiplicities)**.** Let $z\colon[0,1]^n\to[0,1]$ be a mapping with $z(0,\ldots,0)=0$ and z continuous at $(0,\ldots,0)$. Let $M_z\subseteq\mathscr{M}$ be defined by

$$M_z=\{m\in\mathscr{M}\mid\forall e\in\mathscr{E}.\,\mathbf{D}(m,e)\leq z(e(x_1),\ldots,e(x_n))\}$$

We call M_z the *derived set of multiplicities* of z.

Theorem 5.57. *Let $z\colon [0,1]^n \to [0,1]$ be a mapping with $z(0,\ldots,0) = 0$ and z continuous at $(0,\ldots,0)$. Then, an operator $f \in \Sigma$ with $\mathsf{r}(f) = n$ has z as modulus of continuity if*

$$m_f \in M_z$$

with m_f the canonical deterministic denotation of f (Definition 5.54), and M_z the set of derived multiplicities of z (Definition 5.56).

Proof. This is a special case of Theorem 5.65 below. In detail, define $z'\colon \mathscr{E} \to [0,1]$ by $z'(e) = z(e(x_1),\ldots,e(x_n))$. In the proof of Proposition 5.59 below it is shown that z is a modulus of continuity for operator f iff z' is a modulus of continuity for term $f(x_1,\ldots,x_{\mathsf{r}(f)})$ (Definition 5.58). Theorem 5.65 shows that term $f(x_1,\ldots,x_{\mathsf{r}(f)})$ has z' as modulus of continuity if the multiplicity $m_{f(x_1,\ldots,x_{\mathbf{r}(f)})}$ induced by $f(x_1,\ldots,x_{\mathsf{r}(f)})$ (Definition 5.62) is such that $m_{f(x_1,\ldots,x_{\mathbf{r}(f)})} \in M_{z'}$, where $M_{z'} = \{m \in \mathscr{M} \mid \forall e \in \mathscr{E}.\, \mathbf{D}(m,e) \leq z'(e)\}$. Finally, $m_f = m_{f(x_1,\ldots,x_{\mathbf{r}(f)})}$ and $M_z = M_{z'}$. □

5.4.2 Compositional contexts

We proceed by analyzing when a context, i.e. an open term, is continuous. This generalizes the results of the former section whereby the analyzed operator f of the former section corresponds to the context $f(x_1\ldots,x_{\mathsf{r}(f)})$ in this section.

Definition 5.58 (Uniformly continuous term)**.** Let $t \in \mathbb{T}(\Sigma)$ be any open term. A mapping $z\colon \mathscr{E} \to [0,1]$ is an *upper bound on the distance between closed instances of t* (for short an upper bound for t) if

$$\mathbf{d}(\sigma_1(t),\sigma_2(t)) \leq z(\mathbf{d}(\sigma_1,\sigma_2))$$

for all closed substitutions $\sigma_1,\sigma_2\colon \mathscr{V} \to \mathsf{T}(\Sigma)$. An upper bound z for t is a *modulus of continuity of t* if

- z is continuous at $0 \in \mathscr{E}$, i.e. for each $\epsilon > 0$ there exists some $\delta > 0$ s.t. for all $e \in \mathscr{E}$ we get $d_{\mathscr{E}}(e,0) < \delta$ implies $|z(e) - z(0)| < \epsilon$, and
- $z(0) = 0$.

A term t is *uniformly continuous* if t admits some modulus of continuity.

The following proposition relates uniform continuity of operators as discussed in the former section to uniform continuity of terms.

Proposition 5.59. *An operator f is uniformly continuous iff the term $f(x_1,\ldots,x_{\mathsf{r}(f)})$ is uniformly continuous.*

Proof. First, assume that the operator f is uniformly continuous (Definition 4.31). Let $z\colon [0,1]^{\mathsf{r}(f)} \to [0,1]$ be any modulus of continuity for f. We define the mapping $z'\colon \mathscr{E} \to [0,1]$ by $z'(e) = z(e(x_1),\ldots,e(x_{\mathsf{r}(f)}))$ for all $e \in \mathscr{E}$. Now we will show that z' is a modulus of continuity for $f(x_1,\ldots,x_{\mathsf{r}(f)})$, from which the thesis follows. Property $z'(0) = 0$ follows by $z'(0) = z(0(x_1),\ldots,0(x_{\mathsf{r}(f)})) = z(0,\ldots,0) = 0$. It remains to prove continuity at $0 \in \mathscr{E}$. For an arbitrary $\epsilon > 0$, we have to provide a $\delta > 0$ such that $d_{\mathscr{E}}(e,0) < \delta$ implies $z'(e) < \epsilon$ for all $e \in \mathscr{E}$. By the continuity of z at $(0,\ldots,0)$, there exist $\delta_1,\ldots,\delta_{\mathsf{r}(f)}$ such that

$z(v_1,\ldots,v_{\mathsf{r}(f)}) < \epsilon$ whenever $v_i < \delta_i$ for $i = 1,\ldots,\mathsf{r}(f)$. Let $\delta' = \min_{i=1}^{\mathsf{r}(f)} \delta_i$. Assume any process distance $e \in \mathscr{E}$ with $d_{\mathscr{E}}(0,e) < \delta'$, i.e. $e(x_i) < \delta' \leq \delta_i$ for all $i = 1,\ldots,\mathsf{r}(f)$. Then $z'(e) = z(e(x_1),\ldots,e(x_{\mathsf{r}(f)})) < \epsilon$ by continuity of z at $(0,\ldots,0)$. Hence z' is continuous at $0 \in \mathscr{E}$.

Now, assume that the term $f(x_1,\ldots,x_{\mathsf{r}(f)})$ is uniformly continuous. Let $z' : \mathscr{E} \to [0,1]$ be a modulus of continuity for $f(x_1,\ldots,x_{\mathsf{r}(f)})$. Define the mapping $z : [0,1]^{\mathsf{r}(f)} \to [0,1]$ by $z(v_1,\ldots,v_{\mathsf{r}(f)}) = \inf_{v_i = e(x_i) \text{ for } i=1,\ldots,\mathsf{r}(f)} {}^{e\in\mathscr{E}} z'(e)$. We show that z is a modulus of continuity for f, from which the thesis follows. Property $z(0,\ldots,0) = 0$ follows by $z(0,\ldots,0) = z'(0) = 0$. It remains to prove continuity at $(0,\ldots,0)$. For an arbitrary $\epsilon > 0$, we must provide values $\delta_1,\ldots,\delta_{\mathsf{r}(f)}$ such that $z(v_1,\ldots,v_{\mathsf{r}(f)}) < \epsilon$ whenever $v_i < \delta_i$ for $i = 1,\ldots,\mathsf{r}(f)$. By the continuity of z' at $0 \in \mathscr{E}$ there exists a $\delta > 0$ such that $d_{\mathscr{E}}(e,0) < \delta$ implies $z'(e) < \epsilon$ for all $e \in \mathscr{E}$. Let $\delta_i = \delta$ for $i = 1,\ldots,\mathsf{r}(f)$. Assume arbitrary value $v_i < \delta_i = \delta$, for $i = 1,\ldots,\mathsf{r}(f)$. We get $z(v_1,\ldots,v_{\mathsf{r}(f)}) = \inf_{e(x_i) = v_i \text{ for } i=1,\ldots,\mathsf{r}(f)} {}^{e\in\mathscr{E}} z'(e) \leq \inf_{e(x_i) < \delta \text{ for } i=1,\ldots,\mathsf{r}(f)} {}^{e\in\mathscr{E}} z'(e) < \epsilon$. Hence z is continuous at $(0,\ldots,0)$. □

Definition 5.60 (Denotation induced upper bound)**.** For an open term $t \in \mathbb{T}(\Sigma)$, the *upper bound induced by the canonical denotation of* t is the mapping $z_t : \mathscr{E} \to [0,1]$ defined by

$$z_t(e) = \mathbf{A}([\![t]\!], e)$$

for all $e \in \mathscr{E}$.

Notice that the denotation induced upper bound for operator f (Definition 5.47) and the denotation induced upper bound for term $f(x_1,\ldots,x_{\mathsf{r}(f)})$ (Definition 5.60) are related by $z_f(e(x_1),\ldots,e(x_{\mathsf{r}(f)})) = z_{f(x_1,\ldots,x_{\mathsf{r}(f)})}(e)$ for all $e \in \mathscr{E}$. Property $\mathbf{d} \preceq [\![\cdot]\!]$ (Theorem 5.45) gives directly that z_t is an upper bound for t.

It is well-known from analysis that continuous functions are closed under composition. For a given term $f(t_1,\ldots,t_{\mathsf{r}(f)})$, the composition of the modulus of continuity of f with the moduli of continuity of t_i is a modulus of continuity of $f(t_1,\ldots,t_{\mathsf{r}(f)})$.

Theorem 5.61 (Compositionality of moduli of continuity)**.** *Let $t \in \mathbb{T}(\Sigma)$ be any open term. We define $\overline{z}_t : \mathscr{E} \to [0,1]$ by*

$$\overline{z}_t(e) = \begin{cases} e(x) & \text{if } t = x \\ z_f(\overline{z}_{t_1}(e),\ldots,\overline{z}_{t_{\mathsf{r}(f)}}(e)) & \text{if } t = f(t_1,\ldots,t_{\mathsf{r}(f)}). \end{cases}$$

Then it holds that:

1. $\overline{z}_t(0) = 0$*;*
2. *if z_f is continuous at $(0,\ldots,0)$ for all f occurring in t, then $\overline{z}_t$ is continuous at $0 \in \mathscr{E}$;*
3. $\overline{z}_t$ *is an upper bound for* t.

Proof. We prove the three properties by structural induction over t.

1. The base case $t = x$ is immediate since $\overline{z}_t(0) = 0(x) = 0$. Consider the induction step $t = f(t_1,\ldots,t_{\mathsf{r}(f)})$. By definition we have $\overline{z}_t(0) = z_f(\overline{z}_{t_1}(0),\ldots,\overline{z}_{t_{\mathsf{r}(f)}}(0))$. Then, by the inductive hypothesis we have $\overline{z}_{t_i}(0) = 0$ for $i = 1,\ldots,\mathsf{r}(f)$, which gives $z_f(\overline{z}_{t_1}(0),\ldots,\overline{z}_{t_{\mathsf{r}(f)}}(0)) = z_f(0,\ldots,0)$. Finally, we get $z_f(0,\ldots,0) = 0$ from $z_f(0,\ldots,0) = \mathbf{A}([\![f(x_1,\ldots,x_{\mathsf{r}(f)})]\!],0)$ and $\mathbf{A}([\![f(x_1,\ldots,x_{\mathsf{r}(f)})]\!],0) = 0$.

2. The base case $t = x$ is immediate. Consider the inductive step $t = f(t_1, \ldots, t_{\mathsf{r}(f)})$. By definition we have $\overline{z}_t(e) = z_f(\overline{z}_{t_1}(e), \ldots, \overline{z}_{t_{\mathsf{r}(f)}}(e))$. Continuity of $\overline{z}_t$ follows since all $\overline{z}_{t_i}(e)$ are continuous at $0 \in \mathscr{E}$ by the inductive hypothesis, z_f is continuous at $(0, \ldots, 0)$ by the hypothesis, and the composition of continuous functions is continuous.

3. We need to show that $\mathbf{d}(\sigma_1(t), \sigma_2(t)) \leq \overline{z}_t(\mathbf{d}(\sigma_1, \sigma_2))$ for all closed substitutions $\sigma_1, \sigma_2 \colon \mathscr{V} \to \mathsf{T}(\Sigma)$. Consider the base case $t = x$. By definition $\mathbf{d}(\sigma_1, \sigma_2)(x) = \mathbf{d}(\sigma_1(x), \sigma_2(x))$. Then we get $\mathbf{d}(\sigma_1(t), \sigma_2(t)) = \mathbf{d}(\sigma_1(x), \sigma_2(x)) = \mathbf{d}(\sigma_1, \sigma_2)(x) = \overline{z}_t(\mathbf{d}(\sigma_1, \sigma_2))$. Consider the induction step $t = f(t_1, \ldots, t_{\mathsf{r}(f)})$. Then we have $\mathbf{d}(\sigma_1(t), \sigma_2(t)) \leq z_f(\mathbf{d}(\sigma_1(t_1), \sigma_2(t_1)), \ldots, \mathbf{d}(\sigma_1(t_{\mathsf{r}(f)}), \sigma_2(t_{\mathsf{r}(f)}))$ since z_f is an upper bound for f-composed terms. By the induction hypothesis $\mathbf{d}(\sigma_1(t_i), \sigma_2(t_i)) \leq \overline{z}_{t_i}(\mathbf{d}(\sigma_1, \sigma_2))$ and monotonicity of z_f (Proposition 5.26), we derive now the final thesis that $z_f(\mathbf{d}(\sigma_1(t_1), \sigma_2(t_1)), \ldots, \mathbf{d}(\sigma_1(t_{\mathsf{r}(f)}), \sigma_2(t_{\mathsf{r}(f)})))$ is less than or equal to the expression $z_f(\overline{z}_{t_1}(\mathbf{d}(\sigma_1, \sigma_2)), \ldots, \overline{z}_{t_{\mathsf{r}(f)}}(\mathbf{d}(\sigma_1, \sigma_2)))$ which is in fact equal to $\overline{z}_t(\mathbf{d}(\sigma_1, \sigma_2))$.

□

In fact, Theorem 5.61 allows us to use the moduli of continuity of operators (Section 5.4.1) to compute a modulus of continuity of any open term. Now we can show that an term is uniformly continuous if its canonical denotation is definite.

Definition 5.62 (Canonical deterministic denotation of terms)**.** For a term $t \in \mathbb{T}(\Sigma)$ we define the *canonical deterministic denotation of term* t as

$$m_t = \overline{\sup[\![t]\!]}.$$

Notice that we have $m_f = m_{f(x_1, \ldots, x_{\mathsf{r}(f)})}$ for m_f the canonical deterministic denotation of operator f (Definition 5.54). A term t is uniformly continuous if m_t is finite.

Theorem 5.63 (Uniformly continuous term)**.** *Let $t \in \mathbb{T}(\Sigma)$ be an open term. If the canonical deterministic denotation m_t of t is finite then t is uniformly continuous.*

Proof. We will show that $z(e) = \mathbf{D}(m_t, e)$ is a modulus of continuity of t. First, observe that $z(0) = 0$. Moreover, z is continuous at $0 \in \mathscr{E}$ by Proposition 5.51. It remains to show that $z_t(e) \leq z(e)$. We reason by $z_t(e) = \mathbf{A}([\![t]\!], e) = \sup_{P \in [\![t]\!]} \mathbf{P}(P, e) \leq \mathbf{P}(\sup_{P \in [\![t]\!]}, e) \leq \mathbf{D}(\overline{\sup[\![t]\!]}, e) = \mathbf{D}(m_t, e) = z(e)$ with step 3 by monotonicity of $\mathbf{P}$ (Proposition 5.19) and step 4 by Proposition 5.53. □

Definition 5.64 (Distance bound induced multiplicities)**.** Let $z \colon \mathscr{E} \to [0, 1]$ be a mapping with $z(0) = 0$ and z continuous at $0 \in \mathscr{E}$. Let $M_z \subseteq \mathscr{M}$ be the set defined by

$$M_z = \{m \in \mathscr{M} \mid \forall e \in \mathscr{E}. \mathbf{D}(m, e) \leq z(e)\}$$

We call M_z the *multiplicities induced by distance bound* z.

Notice that for all mappings $z' \colon \mathscr{E} \to [0, 1]$ and $z \colon [0, 1]^{\mathsf{r}(f)} \to [0, 1]$ such that for all $e \in \mathscr{E}$ it holds $z'(e) = z(e(x_1), \ldots, e(x_{\mathsf{r}(f)}))$, we have that the derived set of multiplicites $M_{z'}$ of z' (Definition 5.64) and the derived set of multiplicites M_z of z (Definition 5.56) coincide.

Theorem 5.65. *Let $z\colon \mathscr{E} \to [0,1]$ be a mapping with $z(0) = 0$ and z continuous at $0 \in \mathscr{E}$. Then, an open term $t \in \mathbb{T}(\Sigma)$ has z as modulus of continuity if*

$$m_t \in M_z$$

with m_t the canonical deterministic denotation of t (Definition 5.62), and and M_z the multiplicities induced by z (Definition 5.64).

Proof. Let $m_t \in M_z$. First we show that the term t has z as an upper bound. Then, we conclude that z is a modulus of continuity of t.

Let σ_1, σ_2 be any closed substitutions. Then:

$$\begin{aligned}
&\mathbf{d}(\sigma_1(t), \sigma_2(t)) \\
\leq\ &\mathbf{A}([\![t]\!], \mathbf{d}(\sigma_1, \sigma_2)) && \text{(Theorem 5.45)} \\
\leq\ &\mathbf{P}(\sup[\![t]\!], \mathbf{d}(\sigma_1, \sigma_2)) && \text{(Proposition 5.26)} \\
\leq\ &\mathbf{D}(\overline{\sup[\![t]\!]}, \mathbf{d}(\sigma_1, \sigma_2)) && \text{(Proposition 5.53)} \\
=\ &\mathbf{D}(m_t, \mathbf{d}(\sigma_1, \sigma_2)) && \text{(Definition of } m_t\text{)} \\
\leq\ &z(\mathbf{d}(\sigma_1, \sigma_2)) && (m_t \in M_z)
\end{aligned}$$

Hence, z is an upper bound on the distance between instances of t.

Since $z(0) = 0$ and z is continuous at 0, the function z is not only an upper bound on t but a modulus of continuity on t. □

To conclude, the methods provided in Sections 5.2 and 5.3 to compute an upper bound on the distance between closed instances of the term $f(x_1, \ldots, x_{\mathrm{r}(f)})$ can be used to derive the individual compositionality property of the operator f given by its modulus of continuity z_f. Note that z_f depends not only on the rules specifying operator f but also on all those rules which define operators of those processes to which an instance of $f(x_1, \ldots, x_{\mathrm{r}(f)})$ may evolve to. Traditional rule formats define syntactic criteria on single rules in order to guarantee a desired compositionality property of the specified operator. In contrast, our approach derives the compositionality property of an operator from the syntactic properties of those rules which define the operational behavior of processes composed by that operator.

5.5 Closing remarks

We developed a denotational model of open nondeterministic probabilistic processes that captures the basic ingredients that determine the distance between processes: Replication of processes (captured by multiplicities, Definition 5.1), probabilistic choice (captured by probabilistic multiplicities, Definition 5.12), and nondeterministic choice (captured by nondeterministic probabilistic multiplicities, Definition 5.20). Given any open term denoting an open nondeterministic probabilistic process the denotation allows us to compute now an upper bound on the distance between closed instances of that term (Theorems 5.23 and 5.45).

These results together allow us to decide for any given PTSS which operators allow for compositional metric reasoning, i.e. which operators are uniformly continuous (Theorem 5.55). In line with the results of the former chapters uniformly continuous operators are operators that may replicate their composed processes only finitely many times. This translates in the denotational model as sufficient condition that operators with finite canonical deterministic denotations are uniformly continuous.

Our method allows also to compute for any given PTSS a modulus of continuity of each uniformly continuous operator (Definitions 5.47 and 5.54, Theorems 5.45 and 5.55). In reverse, for any given modulus of continuity (semantical specification of the compositionality property of an operator) we provide a method to decide if an operator satisfies this modulus of continuity (Theorem 5.57). In essence, the modulus of continuity induces a maximal denotation (syntactical specification of the compositionality property of that operator) which captures how many times processes combined by that operator may be replicated during their evolution. Any operator with a denotation that does not exceed the induce maximal denotation admits the given modulus of continuity.

The composition of those results allows us then to derive a modulus of continuity of terms by mimicking the syntactic composition of operators (Theorems 5.61 and 5.63). In line with the compositionality results for operators we can derive also for any given modulus of continuity a maximal multiplicity s.t. any term with a lower canonical multiplicity admits that modulus of continuity (Theorem 5.64).

The distance between f-composed processes derived from the canonical denotation of operator f (Definition 5.47) is an upper bound (soundness of the induced distance). However, the distance may not be optimal in the following sense. We call the upper bound z_f induced by the canonical denotation of operator f optimal if for each distances $(\epsilon_1, \ldots, \epsilon_{\mathsf{r}(f)}) \in [0,1]^{\mathsf{r}(f)}$ there are for all $i = 1, \ldots, \mathsf{r}(f)$ terms $s_i, t_i \in \mathsf{T}(\Sigma)$ with $\mathbf{d}(s_i, t_i) = \epsilon_i$ such that $\mathbf{d}(f(s_1, \ldots, s_{\mathsf{r}(f)}), f(t_1, \ldots, t_{\mathsf{r}(f)})) = z_f(\epsilon_1, \ldots, \epsilon_{\mathsf{r}(f)})$. Initial research suggest that the specification of f needs to meet at least the following requirements: no junk rules [AFV01b, Section 5.4.2], f is only finitely recursing (i.e. creates only finitely many copies of its arguments), and only finitely many actions A. The detailed analysis and verification of those requirements will be left as future work.

Another interesting direction is to investigate how the compositional computation of bisimulation distances on Markov Decision Processes [Bac+13] relates to our framework. Asides the obvious difference that [Bac+13] does not consider nondeterministic choice and each process can execute each action, the composition operators defined in [Bac+13, Definition 4] are in our context SOS rules with premises $x_i \xrightarrow{a_i} \mu_i$ and the composition semantics is defined by the rule target (e.g. [Bac+13, Examples 5 & 6]). The compositionality property of 1-safe operators ([Bac+13, Definition 10]) is in fact non-expansiveness and translates to our framework as denotation $\bigoplus_{i=1}^{\mathsf{r}(f)} 1_{x_i}$, while ∞-safe operators are non-extensive operators with denotation $\bigcup_{i=1}^{\mathsf{r}(f)} 1_{x_i}$.

Similar to the former chapters also the results in this chapter should be reinvestigated for other behavioral metrics, such as convex bisimulation metric [Alf+07; Alf+08; Ram10; Mio14], generalized bisimulation metric [Cha+14], trace metrics [AFS04; Bac+15] and metrics based on testing semantics. Finally, we suggest to investigate how the denotational approach to derive specification requirements (developed in this chapter) relates to the logical approach to derive rule formats (developed in [BFG04; GF10]). For this com-

parision the bialgebraic framework seems suitable. Initial ideas can be found at [Kli05; Kli09; Kli10; KS13].

Chapter 6

Axiomatizing bisimulation equivalences and metrics

6.1 Introduction

In the former section we studied the compositionality of probabilistic programming languages both from the operational and from the denotational semantics point of view. We derived general properties on operators by inspecting the specification rules that define the semantics of the operators. However, there are properties that are better understood from an axiomatic point of view, by regarding the language as a signature equipped with an equational theory (see e.g. [Mil89; BBR10]). This is a different way to understand the language that brings new insights on the behavior of its operators and processes. General properties, such as associativity, distributivity, or reduction to basic operators, or specific ones, can be easily derived with equational reasoning, which is also used for the verification of systems.

In [ABV94], Aceto, Bloom & Vaandrager link the operational and the axiomatic approach by providing an algorithm to derive an equational theory for any language whose semantics is defined in terms of SOS rules that meet the GSOS format [BIM95]. This equational theory is sound and ground-complete for bisimulation equivalence [Mil89]. For recent work in that area we refer the interested reader to [Ace+13; GF13] and references therein.

The above mentioned results are developed in the setting of traditional nondeterministic semantics. Using the recently developed equational theory for probabilistic languages [BS01; Hen12] we will lift in this chapter the result of [ABV94] to languages with probabilistic operations. The input of our algorithm is a PGSOS system (cf. Section 2.4) and the output is a sound and ground-complete equational theory for bisimilarity equivalence (cf. Section 2.3.1). The novelty in our approach is to employ many-sorted algebras to axiomatize separately non-deterministic choice, probabilistic choice and their interaction. The main contributions of this chapter are:

1. We generalize the PGSOS format to two-sorted signatures in order to syntactically denote states and distributions (Section 6.2). By doing so, operators can now

be parameterized also on distributions, and moreover, we can neatly express open terms in the rules of the PTSS. While the syntax somehow resembles the alternating model of probabilistic processes [Han94; HJ94], we continue the research line of the former chapters and let PTSS have probabilistic nondeterministic transition systems (Definition 2.10) as models. We show that bisimilarity equivalence is a congruence for any operator whose semantics is defined by rules in the generalized PGSOS format (Theorem 6.8).

2. We provide an algorithm that takes a PGSOS system, and produces an equational theory that is sound and ground-complete for bisimilarity equivalence (Figure 6.1). We show ground-completeness for semantically well-founded PGSOS systems (Theorem 6.24), and we indicate how this result can be extended to arbitrary PGSOS.

3. As a by-product we needed to define a two-sorted calculus for finite probabilistic processes equipped with a sound and ground-complete equational theory for bisimilarity equivalence (Table 6.1). This calculus is adapted from [BS01].

4. We provide an equational theory for the basic calculus that captures exactly the notion of bisimilarity metric (Table 6.4). The equational theory is sound (Theorem 6.25) in the sense that, whenever the equality between the distance of two processes and the distance of two other processes (or a particular value) can be calculated with the calculus, it can also be calculated semantically in the probabilistic transition system. We show that it is also ground-complete (Theorem 6.28), i.e. the inverse implication holds for closed terms.

5. Finally, we provide an algorithm to derive a sound and ground-complete equational theory for bisimilarity metric from a given PGSOS system (Theorem 6.29).

The main results of this chapter have been published in [DGL14]. We generalize the earlier publication by allowing now also strictly discounting bisimulation metrics.

6.2 Preliminaries

We will first introduce two-sorted signatures and their respective term algebras, then generalize the PGSOS rule format to the two-sorted setting and show that also the generalized rule format satisfies the important congruence property.

6.2.1 Many-sorted signatures and term algebras

Let $S = \{s, d\}$ be a set denoting two sorts. Elements of sort $s \in S$ will represent states in the transition system, and elements of sort $d \in S$ will represent distributions over states. We let σ range over the sorts in S. We write S-sorted families X as pairs (X_s, X_d) with the first element X_s denoting the member of sort s and the second element X_d denoting the member of sort d.

An *S-sorted signature* is a structure (F, ar), where (i) F is a set of *function names*, and (ii) $\mathsf{ar}\colon F \to (S^* \times S)$ is the *arity function*. The rank of $f \in F$ is the number of arguments of f, defined by $\mathsf{r}(f) = n$ if $\mathsf{ar}(f) = \sigma_1 \dots \sigma_n \to \sigma$. (We write "$\sigma_1 \dots \sigma_n \to \sigma$" instead of

"$(\sigma_1 \dots \sigma_n, \sigma)$" to highlight that function f maps to sort σ.) Function f is a *constant* if $\mathsf{r}(f) = 0$. To simplify the presentation we will write an S-sorted signature (F, ar) as a pair of disjoint signatures (Σ_s, Σ_d) where Σ_s is the set of operators that map to s and Σ_d is the set of operators that map to d.

Let $\mathcal{V}_s$ and $\mathcal{V}_d$ be two infinite sets of S-sorted variables where $\mathcal{V}_s, \mathcal{V}_d, F$ are all mutually disjoint. We use x, y, z (with possible sub- or super-scripts) to range over $\mathcal{V}_s$, μ, ν to range over $\mathcal{V}_d$ and ζ to range over $\mathcal{V}_s \cup \mathcal{V}_d$.

Definition 6.1. Let Σ_s and Σ_d be two signatures as before and let $V_s \subseteq \mathcal{V}_s$ and $V_d \subseteq \mathcal{V}_d$. We simultaneously define the sets of *state terms* $T(\Sigma_s, V_s, V_d)$ and *distribution terms* $T(\Sigma_d, V_s, V_d)$ as the smallest sets satisfying:

(i) $V_s \subseteq T(\Sigma_s, V_s, V_d)$;

(ii) $V_d \subseteq T(\Sigma_d, V_s, V_d)$;

(iii) $f(\xi_1, \cdots, \xi_{\mathsf{r}(f)}) \in T(\Sigma_\sigma, V_s, V_d)$, if $\mathsf{ar}(f) = \sigma_1 \dots \sigma_n \to \sigma$ and $\xi_i \in T(\Sigma_{\sigma_i}, V_s, V_d)$.

Let $\mathbb{T}(\Sigma) = T(\Sigma_s, \mathcal{V}_s, \mathcal{V}_d)$ denote the set of *open state terms*, $\mathbb{DT}(\Sigma) = T(\Sigma_d, \mathcal{V}_s, \mathcal{V}_d)$ the set of *open distribution terms*, and $\mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma) = T(\Sigma_s, \mathcal{V}_s, \mathcal{V}_d) \cup T(\Sigma_d, \mathcal{V}_s, \mathcal{V}_d)$ the set of all *open terms*. Similarly, we let $\mathsf{T}(\Sigma) = T(\Sigma_s, \emptyset, \emptyset)$, $\mathsf{DT}(\Sigma) = T(\Sigma_d, \emptyset, \emptyset)$, $\mathsf{T}(\Sigma) = T(\Sigma_s, \emptyset, \emptyset) \cup T(\Sigma_d, \emptyset, \emptyset)$ denote the set of all *closed state terms*, *closed distribution terms*, and *closed terms*, respectively. We let t, t', t_1,... range over state terms, θ, θ', θ_1,... range over distribution terms, and ξ, ξ', ξ_1,... range over any kind of terms. With $\mathcal{V}(\xi) \subseteq \mathcal{V}_s \cup \mathcal{V}_d$ we denote the set of variables occurring in term ξ.

Let $\Delta(\mathsf{T}(\Sigma))$ denote the set of all (discrete) probability distributions on $\mathsf{T}(\Sigma)$. We let π range over $\Delta(\mathsf{T}(\Sigma))$. For each $t \in \mathsf{T}(\Sigma)$, let[1] $\delta_t \in \Delta(\mathsf{T}(\Sigma))$ denote the *Dirac distribution*, i.e., $\delta_t(t) = 1$ and $\delta_t(t') = 0$ if t and t' are not syntactically equal. For $X \subseteq \mathsf{T}(\Sigma)$ we define $\pi(X) = \sum_{t \in X} \pi(t)$. The convex combination $\sum_{i \in I} p_i \pi_i$ of a family $\{\pi_i\}_{i \in I}$ of probability distributions with $p_i \in (0, 1]$ and $\sum_{i \in I} p_i = 1$ is defined by $(\sum_{i \in I} p_i \pi_i)(t) = \sum_{i \in I} (p_i \pi_i(t))$.

The type of signatures we consider has a particular construction. We start from a signature Σ_s of functions mapping into sort s and construct the signature Σ_d of functions mapping into sort d as follows. For each $f \in F_s$ we include a function symbol $\boldsymbol{f} \in F_d$ with $\mathsf{ar}(\boldsymbol{f}) = d \dots d \to d$ and $\mathsf{r}(\boldsymbol{f}) = \mathsf{r}(f)$. We call $\boldsymbol{f}$ the *probabilistic lifting* of f. (We use boldface fonts to indicate that a function in Σ_d is the probabilistic lifting of another in Σ_s.) Moreover Σ_d includes the following additional operators:

- δ with arity $\mathsf{ar}(\delta) = s \to d$ and
- $\oplus_p$ with $p \in \mathbb{Q} \cap (0, 1)$ and $\mathsf{ar}(\oplus_p) = dd \to d$.

We call S-sorted signatures $\Sigma = (\Sigma_s, \Sigma_d)$ that follow this construction as *probabilistically lifted signatures*.

Operators δ and $\oplus_p$ are used to construct discrete probability functions of countable support: $\delta(t)$ is interpreted as a distribution that assigns probability 1 to the state term

[1] In the former chapters we denoted the Dirac distribution with all probability mass at term t by $\delta(t)$. However, in this chapter $\delta(t)$ will denote a special function in the lifted signature Σ_d. To avoid confusion we will use the alternative notation δ_t for the Dirac distribution.

t and probability 0 to any other term t' (syntactically) different from t, and $\theta_1 \oplus_p \theta_2$ represents a distribution that weights with p the distribution represented by the term θ_1 and with $1-p$ the distribution represented by θ_2. Moreover, a probabilistically lifted operator $\boldsymbol{f}$ is interpreted by lifting the probabilities of the operands to terms composed with the operator f.

Formally, the algebra associated with a probabilistically lifted signature $\Sigma = (\Sigma_s, \Sigma_d)$ is defined as follows. For sort s, it is the freely generated algebraic structure $\mathsf{T}(\Sigma)$. For sort d, it is defined by the carrier $\Delta(\mathsf{T}(\Sigma))$ and the following interpretation:

- $[\![\delta(t)]\!] = \delta_t$ for all $t \in \mathsf{T}(\Sigma)$.
- $[\![\theta_1 \oplus_p \theta_2]\!] = p[\![\theta_1]\!] + (1-p)[\![\theta_2]\!]$ for $\theta_1, \theta_2 \in \mathsf{DT}(\Sigma)$.
- $[\![\boldsymbol{f}(\theta_1, \ldots, \theta_{\mathsf{r}(f)})]\!](f(\xi_1, \ldots, \xi_{\mathsf{r}(f)})) = \begin{cases} \prod_{\sigma_i = s} [\![\theta_i]\!](\xi_i) & \text{if for all } \sigma_j = d, \theta_j = \xi_j \\ 0 & \text{otherwise} \end{cases}$

with $\prod \emptyset = 1$. Notice that in the semantics of a lifted function $\boldsymbol{f}$, the product considers only the distributions related to the s-sorted positions in f, while the distribution terms corresponding to the d-sorted positions in f should be matched exactly to the parameters of $\boldsymbol{f}$.

A *substitution*[2] ρ is a mapping $\mathcal{V}_s \cup \mathcal{V}_d \to \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)$ such that $\rho(x) \in \mathbb{T}(\Sigma)$, for all $x \in \mathcal{V}_s$, and $\rho(\mu) \in \mathbb{DT}(\Sigma)$, for all $\mu \in \mathcal{V}_d$. A substitution is closed if it maps each variable to a closed term. The *interpretation* of open terms with respect to a substitution ρ is defined for state terms as usual by $[\![x]\!]_\rho = \rho(x)$, $[\![f(t_1, \ldots, t_{\mathsf{r}(f)}]\!]_\rho = f([\![t_1]\!]_\rho, \ldots, [\![t_{\mathsf{r}(f)}]\!]_\rho)$, and for distribution terms by

- $[\![\mu]\!]_\rho = \rho(\mu)$,
- $[\![\delta(t)]\!]_\rho = \delta_{[\![t]\!]_\rho}$ for all $t \in \mathsf{T}(\Sigma)$,
- $[\![\theta_1 \oplus_p \theta_2]\!]_\rho = p[\![\theta_1]\!]_\rho + (1-p)[\![\theta_2]\!]_\rho$ for $\theta_1, \theta_2 \in \mathsf{DT}(\Sigma)$, and
- $[\![\boldsymbol{f}(\theta_1, \ldots, \theta_{\mathsf{r}(f)})]\!]_\rho(f(\xi_1, \ldots, \xi_{\mathsf{r}(f)})) = \begin{cases} \prod_{\sigma_i = s} [\![\theta_i]\!]_\rho(\xi_i) & \text{if for all } \sigma_j = d, \theta_j = \xi_j \\ 0 & \text{otherwise} \end{cases}$

We may write $\rho(t)$ for $[\![t]\!]_\rho$, and $\rho(\theta)$ for $[\![\theta]\!]_\rho$.

We remark that lifted operators in distribution terms are distributive w.r.t. $\oplus_p$ at all s-sorted positions, i.e. if $f \in \Sigma_s$ has arity $\mathsf{ar}(f) = \sigma_1 \ldots \sigma_n \to s$ with $\sigma_j = s$, then $\boldsymbol{f} \in \Sigma_d$ is distributive w.r.t. $\oplus_p$ at position j, i.e. $[\![\rho(\boldsymbol{f}(\ldots, \xi_{j-1}, \theta_1 \oplus_p \theta_2, \xi_{j+1}, \ldots))]\!] = [\![\rho(\boldsymbol{f}(\ldots, \xi_{j-1}, \theta_1, \xi_{j+1}, \ldots) \oplus_p \boldsymbol{f}(\ldots, \xi_{j-1}, \theta_2, \xi_{j+1}, \ldots))]\!]$ for any closed substitution ρ. The proof follows directly from the definition of $[\![\,]\!]$. However, notice that $\boldsymbol{f}$ *does not* distribute w.r.t. $\oplus_p$ at a position k with $\sigma_k = d$.

Example 6.2. We define now the signature of probabilistic CSS. Let $\Sigma = (\Sigma_s, \Sigma_d)$ be a probabilistically lifted signature such that Σ_s is the signature containing:

- the constant 0 (stop process) of sort s, i.e., $\mathsf{ar}(0) = s$;

[2] We denote substitutions in this chapter by ρ since the symbol σ denotes in this chapter sorts of S.

- a family of unary probabilistic prefix operators $a.$ with $a \in A$ and $\mathsf{ar}(a) = d \to s$,
- and the binary operator $+$ (alternative composition or sum) with $\mathsf{ar}(+) = ss \to s$.

Moreover, Σ_d contains δ, all binary operators $\oplus_p$ with $p \in \mathbb{Q} \cap (0,1)$, and the lifted operators:

- the constant $\mathbf{0}$ with $\mathsf{ar}(\mathbf{0}) = d$;
- the family of unary operators $\boldsymbol{a}.$, with $a \in A$ and $\mathsf{ar}(\boldsymbol{a}) = d \to d$,
- and the binary operator $+$ with $\mathsf{ar}(+) = dd \to d$.

The intended meaning of the probabilistic prefix operator $a.\theta$ is that this term can perform action a and moves to term t with probability $[\![\theta]\!](t)$. The $+$ operator has the usual meaning.

Notice that the same Dirac distribution of a closed state term can be written in several ways using the operator δ or the probabilistically lifted signature. Thus we have that the terms $\delta(a.(\boldsymbol{a}.\mathbf{0} + \boldsymbol{b}.\mathbf{0}))$, $\boldsymbol{a}.(\delta(a.\mathbf{0} + b.\mathbf{0}))$, and $\boldsymbol{a}.(\boldsymbol{a}.\mathbf{0} + \boldsymbol{b}.\mathbf{0}))$ represent the same distribution. Indeed, it is not difficult to show that $\delta_{a.(\boldsymbol{a}.\mathbf{0}+\boldsymbol{b}.\mathbf{0})} = [\![\delta(a.(\boldsymbol{a}.\mathbf{0} + \boldsymbol{b}.\mathbf{0}))]\!] = [\![\boldsymbol{a}.(\delta(a.\mathbf{0} + b.\mathbf{0}))]\!] = [\![\boldsymbol{a}.(\boldsymbol{a}.\mathbf{0} + \boldsymbol{b}.\mathbf{0}))]\!]$. However, the Dirac operator δ is necessary to construct open distribution terms. The term $\delta(x)$ cannot be written in any other form, since the instance of the state term variable x is not yet known.

6.2.2 Probabilistic transition system specifications

Now we generalize the probabilistic GSOS format as introduced in Section 2.4 by allowing operators of sort s with arguments of sort either sort s or d.

Definition 6.3 (Generalized PGSOS rule)**.** A *generalized PGSOS rule* has the form:

$$\frac{\{x_i \xrightarrow{a_{i,m}} \mu_{i,m} \mid i \in I, m \in M_i\} \quad \{x_i \xrightarrow{b_{i,n}} \not\to \mid i \in I, n \in N_i\}}{f(\zeta_1, \ldots, \zeta_{\mathsf{r}(f)}) \xrightarrow{a} \theta}$$

with $f \in \Sigma_s$ a function symbol, I, M_i, N_i are finite index sets, $a_{i,m}, b_{i,n}, a \in A$ are actions, $x_i \in \mathcal{V}_s$, $\zeta_i \in \mathcal{V}_s \cup \mathcal{V}_d$, $\mu_{i,m} \in \mathcal{V}_d$ are variables, $\theta \in \mathbb{DT}(\Sigma)$ a distribution term, and satisfying the following constraints:

1. all $\mu_{i,m}$ and ζ_j for $i \in I, m \in M_i, j \in \{1, \ldots, \mathsf{r}(f)\}$ are pairwise different;
2. $\{x_i \mid i \in I\} \subseteq \{\zeta_i \mid i = 1, \ldots, \mathsf{r}(f)\}$;
3. $\mathcal{V}(\theta) \subseteq \{\mu_{i,m} \mid i \in I, m \in M_i\} \cup \{\zeta_i \mid i \in I\}$.

A *probabilistic transition system specification* in generalized PGSOS format is a structure $P = (\Sigma, A, R)$ where Σ is a probabilistically lifted signature, A is a finite set of labels and R is a finite set of generalized PGSOS rules. For any rule $r \in R$, literals above the line are called *premises*, notation $\mathrm{prem}(r)$; the literal below the line is called *conclusion*, notation $\mathrm{conc}(r)$. Given a positive literal $t \xrightarrow{a} \theta$ and a closed substitution ρ, $[\![t \xrightarrow{a} \theta]\!]_\rho$

denotes the transition $\rho(t) \xrightarrow{a} [\![\rho(\theta)]\!]$. For negative literals, $[\![t \overset{a}{\not\to}]\!]_\rho$ denotes $\rho(t) \overset{a}{\not\to}$. A *supported model* of P is a PTS $(\mathsf{T}(\Sigma), A, \to)$ satisfying that $t \xrightarrow{a} \pi \in \to$ iff there is a rule $r \in R$ and a substitution ρ such that all premises of r hold, i.e. $\to \models [\![\mathrm{prem}(r)]\!]_\rho$, and the conclusion instantiates to $t \xrightarrow{a} \pi$, i.e. $[\![\mathrm{conc}(r)]\!]_\rho = t \xrightarrow{a} \pi$. Each PTSS has a supported model which is, moreover, unique.

A crucial property of process description languages to ensure compositional modeling is the compatibility of process operators with the behavioral relation. In algebraic terms the compatibility of a behavioral equivalence R with an operator f is expressed by the congruence property which is defined as $f(\xi_1, \ldots, \xi_{\mathsf{r}(f)})\, \mathsf{R}\, f(\xi'_1, \ldots, \xi'_{\mathsf{r}(f)})$ whenever $\xi_i\, \mathsf{R}\, \xi'_i$ with $\xi_i, \xi'_i \in \mathsf{T}(\Sigma)$ if $\sigma_i = s$ and $[\![\xi_i]\!]\, \overline{\mathsf{R}}\, [\![\xi'_i]\!]$ with $\xi_i, \xi'_i \in \mathsf{DT}(\Sigma)$ if $\sigma_i = d$.

Before we can show that bisimilarity equivalence is a congruence for all operators specified by generalized PGSOS rules (Theorem 6.8) we first need define the congruence closure and lifting of relations, and show their respective properties.

Definition 6.4. Let $P = (\Sigma, A, R)$ be a PTSS with Σ a probabilistically lifted signature. The congruence closure of a relation $\mathsf{R} \subseteq (\mathsf{T}(\Sigma) \cup \mathsf{DT}(\Sigma)) \times (\mathsf{T}(\Sigma) \cup \mathsf{DT}(\Sigma))$ is defined as the smallest relation $P(\mathsf{R}) \subseteq (\mathsf{T}(\Sigma) \cup \mathsf{DT}(\Sigma)) \times (\mathsf{T}(\Sigma) \cup \mathsf{DT}(\Sigma))$ such that:

(i) $\mathsf{R} \subseteq P(\mathsf{R})$, and

(ii) $f(\xi_1, \cdots, \xi_{\mathsf{r}(f)})\, P(\mathsf{R})\, f(\xi'_1, \cdots, \xi'_{\mathsf{r}(f)})$ whenever $\xi_i\, P(\mathsf{R})\, \xi'_i$ for all $i = 1, \ldots, \mathsf{r}(f)$

Lemma 6.5. *Let $\rho, \rho' : (\mathcal{V}_s \cup \mathcal{V}_d) \to (T(\Sigma) \cup DT(\Sigma))$ be two closed substitutions with $\rho(\zeta)\ R\ \rho'(\zeta)$ for all $\zeta \in \mathcal{V}_s \cup \mathcal{V}_d$. Then $\rho(\tau)\, P(R)\, \rho'(\tau)$ for all terms $\tau \in \mathbb{T}(\Sigma) \cup \mathbb{DT}(\Sigma)$.*

Proof. Straightforward by structural induction. □

Definition 6.6. Let $P = (\Sigma, A, R)$ be a PTSS with Σ a probabilistically lifted signature. The lifting of $\mathsf{R} \subseteq (\mathsf{T}(\Sigma) \cup \mathsf{DT}(\Sigma)) \times (\mathsf{T}(\Sigma) \cup \mathsf{DT}(\Sigma))$ is defined as the relation $\overline{\mathsf{R}} \subseteq \Delta(\mathsf{T}(\Sigma)) \times \Delta(\mathsf{T}(\Sigma))$ such that $\pi\, \overline{\mathsf{R}}\, \pi'$ iff $\pi(X) = \pi'(X)$ for all $X \subseteq \mathsf{T}(\Sigma)$ that are R-closed(X).

The congruence closure on distribution terms coincides with the lifting of the congruence closure on state terms. This is a fundamental property guaranteeing that earlier results on congruence, non-expansiveness and continuity of [DL12; LGD12; GT13; GT14] carry over to the generalized PGSOS format.

Lemma 6.7. *Let $R \subseteq (T(\Sigma) \cup DT(\Sigma)) \times (T(\Sigma) \cup DT(\Sigma))$ with $[\![\theta]\!]\, \overline{R}\, [\![\theta']\!]$ whenever $\theta\ R\ \theta'$ for all $\theta, \theta' \in DT(\Sigma)$. Then, for any closed distribution terms $\theta, \theta' \in DT(\Sigma)$ with $\theta\, P(R)\, \theta'$ we have $[\![\theta]\!]\, \overline{P(R)}\, [\![\theta']\!]$.*

Proof. First, observe that there is an open distribution term $\vartheta \in \mathbb{DT}(\Sigma)$ and closed substitutions ρ, ρ' with $\rho(x)\, P(\mathsf{R})\, \rho'(x)$ for $x \in \mathcal{V}_s$ and $\rho(\mu)\ \mathsf{R}\ \rho'(\mu)$ for $\mu \in \mathcal{V}_d$ such that $\rho(\vartheta) = \theta$ and $\rho'(\vartheta) = \theta'$. We proceed by induction over ϑ.

Let $\vartheta = \delta(t)$ for $t \in \mathbb{T}(\Sigma)$. Lemma 6.5 gives $\rho(t)\, P(\mathsf{R})\, \rho'(t)$. Hence, $\delta_{\rho(t)}\, \overline{P(\mathsf{R})}\, \delta_{\rho'(t)}$. We conclude this case by observing that $[\![\rho(\delta(t))]\!] = \delta_{\rho(t)}$ and $[\![\rho'(\delta(t))]\!] = \delta_{\rho'(t)}$.

Let $\vartheta = \mu$ for $\mu \in \mathcal{V}_d$. From $\rho(\mu)\ \mathsf{R}\ \rho'(\mu)$ and the assumptions on R we derive $[\![\rho(\mu)]\!]\, \overline{\mathsf{R}}\, [\![\rho'(\mu)]\!]$. Hence, $[\![\rho(\mu)]\!]\, \overline{P(\mathsf{R})}\, [\![\rho'(\mu)]\!]$.

Let $\vartheta = \vartheta_1 \oplus_p \vartheta_2$ with $\rho(\vartheta_i)\, P(\mathsf{R})\, \rho'(\vartheta_i)$ for $i = 1,2$. By induction hypothesis we have for $i = 1,2$ that $[\![\rho(\vartheta_i)]\!]\, \overline{P(\mathsf{R})}\, [\![\rho'(\vartheta_i)]\!]$, i.e. $[\![\rho(\vartheta_i)]\!](X) = [\![\rho'(\vartheta_i)]\!](X)$ for all $X \subseteq \mathsf{T}(\Sigma)$ that are R-closed(X). Hence, $[\![\rho(\vartheta_1 \oplus_p \vartheta_2)]\!](X) = p[\![\rho(\vartheta_1]\!](X) + (1-p)[\![\rho(\vartheta_2)]\!](X) = p[\![\rho'(\vartheta_1]\!](X) + (1-p)[\![\rho'(\vartheta_2)]\!](X) = [\![\rho'(\vartheta_1 \oplus_p \vartheta_2)]\!](X)$. In other words we get now $[\![\rho(\vartheta_1 \oplus_p \vartheta_2)]\!]\, \overline{P(\mathsf{R})}\, [\![\rho'(\vartheta_1 \oplus_p \vartheta_2)]\!]$.

Let $\vartheta = f(\vartheta_1, \ldots, \vartheta_{\mathsf{r}(f)})$ with $\rho(\vartheta_i)\, P(\mathsf{R})\, \rho'(\vartheta_i)$ for $i = 1, \ldots, \mathsf{r}(f)$ with arity of operator f given by $\mathsf{ar}(f) = \sigma_1 \ldots \sigma_n \to \sigma$. The interpretation of the closed distribution term $\rho(\vartheta)$ is $[\![\rho(f(\vartheta_1, \ldots, \vartheta_{\mathsf{r}(f)}))]\!](f(\xi_1, \ldots, \xi_{\mathsf{r}(f)})) = \prod_{\sigma_i = s} [\![\vartheta_i]\!](\xi_i)$ if for all $\sigma_j = d$ we have ϑ_j and ξ_j are syntactically equal, otherwise 0; and in the same matter also the interpretation of $\rho'(f(\vartheta_1, \ldots, \vartheta_{\mathsf{r}(f)}))$. Let X be $P(\mathsf{R})$-closed(X). Then, $f^{-1}(X) = \prod_{i=1}^{\mathsf{r}(f)} X_i$ for some $X_i \subseteq \mathsf{T}(\Sigma)$ if $\sigma_i = s$ and $X_i \subseteq \mathsf{DT}(\Sigma)$ if $\sigma_i = d$. It is clear that $P(\mathsf{R})$-closed(X_i). By the induction hypothesis we get $[\![\rho(\vartheta_i)]\!]\, \overline{P(\mathsf{R})}\, [\![\rho'(\vartheta_i)]\!]$, i.e. $[\![\rho(\vartheta_i)]\!](X_i) = [\![\rho'(\vartheta_i)]\!](X_i)$ if $\sigma_i = s$; and $\rho(\vartheta_i)\, \overline{P(\mathsf{R})}\, \rho'(\vartheta_i)$ if $\sigma_i = d$. By Lemma 6.5 we get $[\![\rho(f(\vartheta_1, \ldots, \vartheta_{\mathsf{r}(f)}))]\!](X) = [\![\rho'(f(\vartheta_1, \ldots, \vartheta_{\mathsf{r}(f)}))]\!](X)$. □

Lemmas 6.5 and 6.7 allow us to prove that bisimilarity equivalence is a congruence for all operators defined in the generalized PGSOS format.

Theorem 6.8. *Let $P = (\Sigma, A, R)$ be a PTSS in generalized PGSOS format. Then, bisimilarity equivalence is a congruence for all operators defined by P.*

Proof. We will show that the congruence closure $P(\sim)$ of the bisimilarity equivalence relation $\sim$ is also a bisimulation. This implies that $\sim$ is a congruence for all operators defined by P.

Let $t_1, t_2 \in \mathsf{T}(\Sigma)$ with $t_1\, P(\sim)\, t_2$. From $t_1\, P(\sim)\, t_2$ we derive that there is a $t \in \mathbb{T}(\Sigma)$ and substitutions ρ_1, ρ_2 with $\rho_1(t) = t_1$, $\rho_2(t) = t_2$ and $\rho_1(x) \sim \rho_2(x)$ for all $x \in \mathcal{V}_s$. We proceed by induction over t and verify that the transfer property is given, i.e. whenever $\rho_1(t) \xrightarrow{a} \pi_1$ then there is a $\rho_2(t) \xrightarrow{a} \pi_2$ with $\pi_1\, \overline{P(\sim)}\, \pi_2$.

The base case $t = x$ is trivial by the fact that $\rho_1(x) \sim \rho_2(x)$. Let $t = f(\xi_1, \ldots, \xi_{\mathsf{r}(f)})$ with $\xi_i \in \mathbb{T}(\Sigma)$ if $\sigma_i = s$ and $\xi_i \in \mathbb{DT}(\Sigma)$ if $\sigma_i = d$. By the induction hypothesis we have $\rho_1(\xi_i)\, P(\sim)\, \rho_2(\xi_i)$. Assume $P \models \rho_1(t) \xrightarrow{a} \pi_1$, i.e. there is a rule $r \in R$ defining operator f such that the transition $\rho_1(t) \xrightarrow{a} \pi_1$ can be derived by a substitution ρ_1' extending ρ_1. Note that $[\![\rho_1'(\mathsf{trgt}(r))]\!] = \pi_1$. Because $\rho_1(\xi_i)\, P(\sim)\, \rho_2(\xi_i)$ we get that ρ_2 can be extended to a substitution ρ_2' such that $P \models \rho_2'(\mathsf{prem}(r))$ and the transition $\rho_2(t) \xrightarrow{a} \pi_2$ with $[\![\rho_2'(\mathsf{trgt}(r))]\!] = \pi_2$ can be derived. By Lemma 6.5 we get $\rho_1'(\mathsf{trgt}(r))\, P(\sim)\, \rho_2'(\mathsf{trgt}(r))$ and by Lemma 6.7 we conclude $[\![\rho_1'(\mathsf{trgt}(r))]\!] = \pi_1\, P(\sim)\, \pi_2 = [\![\rho_2'(\mathsf{trgt}(r))]\!]$. We conclude that the transfer property is given for the congruence closure $P(\sim)$.

We remark that the (common) requirement that the number of rules in a PGSOS TSS is finite is not required for the congruence property. However, if the number of rules is infinite, then the induced PTS may have also infinite branching. □

Example 6.9. Let Σ_{CCS} be the signature defined on the Example 6.2. The PTSS in generalized PGSOS format $P_{CCS} = (\Sigma_{CCS}, A, R)$ is given by the following rules R:

$$\frac{}{a.\mu \xrightarrow{a} \mu} \qquad \frac{x \xrightarrow{a} \mu}{x + y \xrightarrow{a} \mu} \qquad \frac{y \xrightarrow{a} \mu}{x + y \xrightarrow{a} \mu} \tag{6.1}$$

6.3 Axiomatization of bisimilarity equivalence

The technique to derive an axiomatization for PGSOS operators follows the same strategy as in [ABV94]. It starts with a given axiomatization of a basic calculus which is a probabilistic extension of CCS similar to [BS01]. Then, according to the rules, axioms are provided for any other operator so that these operators can be eliminated in the sense that every closed term can be equated to another closed term in the basic calculus. To introduce these new axioms, operators are split in three classes: distinctive, smooth, and non-smooth. *Distinctive operators* are well-behaved operators that distribute with summation and the probabilistic operators $\oplus_p$ and δ. Moreover, each of its defining rules can be directly mapped into axioms. *Smooth operators* are a generalization of distinctive operators in the sense that the set of rules defining the semantics of a smooth operator can be split in disjoint sets, each one of them satisfying the conditions of distinctive operators. Thus a smooth operator can be represented as a non-deterministic sum of distinctive operators. For each *non-smooth operator*, a new smooth operator is introduced that, when properly instantiated, shows the same behavior as the original term with the non-smooth operator. Precisely the equality between these terms is introduce as a new axiom.

This section introduces these results and provides an algorithm that, given a PTSS P in generalized PGSOS format, generates an equational theory for all operators in P that is sound and ground-complete for bisimilarity equivalence.

6.3.1 Axiomatizing finite probabilistic trees

We will use $\bigoplus_{i\in\{1..n\}} p_i\theta_i$ as a shorthand for $\theta_1 \oplus_{\frac{p_1}{\sum_{j=1}^{n} p_j}} (\theta_2 \oplus_{\frac{p_2}{\sum_{j=2}^{n} p_j}} (\cdots(\theta_{n-1} \oplus_{\frac{p_{n-1}}{\sum_{j=n-1}^{n} p_j}} \theta_n)\cdots))$, and $\sum_{i\in\{1..n\}} t_i$ as a shorthand for $t_1 + \cdots + t_n$.

Definition 6.10. A closed term $t \in \mathsf{T}(\Sigma)$ is in *normal form* if either $t = 0$ or $t = \sum_{i\in I} a_i.\theta_i$ with $\theta_i \in \mathsf{DT}(\Sigma)$ in normal form. A closed term $\theta \in \mathsf{DT}(\Sigma)$ is in *normal form* if $\theta = \bigoplus_{i\in I} p_i\delta(t_i)$, with $t_i \in \mathsf{T}(\Sigma)$ in normal form.

Let E_{CCS} be the set of equations of Table 6.1. The axioms N1–N4 are standard for nondeterministic choice of reactive systems [Mil89]. The axioms P1–P3 are standard for probabilistic choice [BS01]. Moreover, axioms NP1–NP5 allow to normalize distribution terms similar to the normalization of state terms by axioms N1–N4. The axiomatization of [BS01] did not require those axioms because distribution terms were assumed to be already in normal form.

Equational reasoning over multi-sorted algebras [GM82] requires non-empty carrier sets. For E_{CCS} and all its following extensions this holds since $0 \in \mathsf{T}(\Sigma)$ and $\delta(0) \in \mathsf{DT}(\Sigma)$. A set of S-sorted equations E over signature Σ is a sound and ground-complete axiomatization of bisimilarity equivalence of P if for all $t, t' \in \mathsf{T}(\Sigma)$, $E \vdash t = t'$ iff $t \sim t'$.

In order to show ground-completeness of E_{CCS} we require that the axiomatization is normalizing for both sort s and d, i.e. that for each closed term $\xi \in \mathsf{T}(\Sigma)$ there is a closed term $\xi' \in \mathsf{T}(\Sigma)$ in normal form such that $E_{CCS} \vdash \xi = \xi'$. The proof of the next lemma follows as usual by transforming the axiom system into a term rewriting system, showing that it is strongly normalizing modulo commutativity and associativity, and that the normal form is indeed in the expected format.

$$x + y = y + x \quad \text{(N1)} \qquad (\mu_1 \oplus_p \mu_2) + \mu_3 = (\mu_1 + \mu_3) \oplus_p (\mu_2 + \mu_3) \quad \text{(NP1)}$$
$$(x + y) + z = x + (y + z) \quad \text{(N2)} \qquad \mu_1 + (\mu_2 \oplus_p \mu_3) = (\mu_1 + \mu_2) \oplus_p (\mu_1 + \mu_3) \quad \text{(NP2)}$$
$$x + 0 = x \quad \text{(N3)} \qquad \delta(x) + \delta(y) = \delta(x + y) \quad \text{(NP3)}$$
$$x + x = x \quad \text{(N4)} \qquad \boldsymbol{a}.\mu = \delta(a.\mu) \quad \text{(NP4)}$$
$$\mu \oplus_p \mu = \mu \quad \text{(P1)} \qquad \mathbf{0} = \delta(0) \quad \text{(NP5)}$$
$$\mu_1 \oplus_p \mu_2 = \mu_2 \oplus_{1-p} \mu_1 \quad \text{(P2)} \qquad \mu_1 \oplus_{p_1} (\mu_2 \oplus_{\frac{p_2}{1-p_1}} \mu_3) = (\mu_1 \oplus_{\frac{p_1}{p_1+p_2}} \mu_2) \oplus_{p_1+p_2} \mu_3 \quad \text{(P3)}$$

Table 6.1: Axiomatization of bisimilarity equivalence of CCS.

Lemma 6.11. *The axiom system E_{CCS} is normalizing.*

Proof. To prove this result we use an appropriate term rewriting systems. We will define rewriting rules based on the axioms set and a weight function over terms. In addition, because nondeterministic choice and probabilistic choice satisfies commutativity and associativity, we work with terms modulo axioms (N1), (N2), (P2) and (P3). A term τ represents the set of terms that can be equated to τ by these axioms.

The rewriting rules are obtained from axioms (N3), (N4), (P1), (NP1), (NP2), (NP3) (NP4) and (NP5) by directing them from left to right. For example, (N3) generates the rewriting rule $x + 0 \rightsquigarrow x$. The weight function $w : \mathsf{T}(\Sigma) \to \mathbb{N}$ is defined by:

$$w(0) = 1 \qquad w(\mathbf{0}) = 3$$
$$w(a.\theta) = w(\theta) \qquad w(\boldsymbol{a}.\theta) = w(\theta) + 2$$
$$w(t + t') = 3 \cdot w(t) \cdot w(t') + 1 \qquad w(\theta + \theta') = 3 \cdot w(\theta) \cdot w(\theta') + 1$$
$$w(\delta(t)) = w(t) + 1 \qquad w(\theta \oplus_p \theta') = w(\theta) + w(\theta') + 1$$

Each application of a rewrite rule strictly decreases the weight of a term. Hence the rewriting process terminates. When no rewrite rule can be applied, the term can be shown to be in head normal form. □

Example 6.12. We normalize the closed term $a.(\boldsymbol{a}.((\boldsymbol{b}.\mathbf{0} \oplus_p \boldsymbol{c}.\mathbf{0}) + \delta(c.\mathbf{0})))$

$$\begin{aligned}
& a.(\boldsymbol{a}.((\boldsymbol{b}.\mathbf{0} \oplus_p \boldsymbol{c}.\mathbf{0}) + \delta(c.\mathbf{0}))) \\
\rightsquigarrow\ & a.((\delta(a.(\boldsymbol{b}.\mathbf{0} \oplus_p \boldsymbol{c}.\mathbf{0}) + \delta(c.\mathbf{0})))) \\
\rightsquigarrow\ & a.((\delta(a.(\delta(b.\mathbf{0}) \oplus_p \delta(c.\mathbf{0})) + \delta(c.\mathbf{0})))) \\
\rightsquigarrow\ & a.(\delta(a.((\delta(b.\mathbf{0}) + \delta(c.\mathbf{0})) \oplus_p (\delta(c.\mathbf{0}) + \delta(c.\mathbf{0}))))) \\
\rightsquigarrow\ & a.(\delta(a.((\delta(b.\mathbf{0}) + \delta(c.\mathbf{0})) \oplus_p \delta(c.\mathbf{0})))) \\
\rightsquigarrow\ & a.(\delta(a.(\delta(b.\mathbf{0} + c.\mathbf{0}) \oplus_p \delta(c.\mathbf{0})))) \\
\rightsquigarrow\ & a.(\delta(a.(\delta(b.\delta(0) + c.\delta(0)) \oplus_p \delta(c.\delta(0)))))
\end{aligned}$$

The proof of soundness for axioms involving state terms follows as usual: for each axiom we find a bisimulation relation that shows that its instances are bisimilar. For axioms on distribution terms we prove that both sides of the equation represent exactly the same distribution. Ground-completeness is proven by first reducing to normal form and then showing that, for two bisimilar state terms in normal form, the transfer properties induce a proof using the axioms. Similarly, two distribution terms in normal form that represent the same distribution up to bisimulation, can be reduce to the same term using the axioms.

Theorem 6.13. *E_{CCS} is sound and ground-complete for bisimilarity equivalence.*

Proof. We start by showing that E_{CCS} is sound for *CCS*. The proof for axioms (N1-N4) proceeds in the same way as in the non-probabilistic context. Axioms (P1-P3) are straightforward by calculation.

The soundness of axiom (NP1) is given by the by the following reasoning steps: Let ρ a closed substitution and $t \in \mathsf{T}(\Sigma)$; if t does not have the form $t_1 + t_2$, then $[\![(\mu_0 \oplus_p \mu_1) + \mu_2]\!]_\rho(t) = 0 = [\![(\mu_0 + \mu_2) \oplus_p (\mu_1 + \mu_2)]\!]_\rho(t)$. On the other hand, if t has form $t_1 + t_2$,

$$\begin{aligned}
& [\![(\mu_0 \oplus_p \mu_1) + \mu_2]\!]_\rho(t_1 + t_2) \\
= \; & [\![(\mu_0 \oplus_p \mu_1)]\!]_\rho(t_1).[\![\mu_2]\!]_\rho(t_2) \\
= \; & (p.[\![\mu_0]\!]_\rho(t_1) + (1-p)[\![\mu_1]\!]_\rho(t_1)).[\![\mu_2]\!]_\rho(t_2) \\
= \; & p.[\![\mu_0]\!]_\rho(t_1).[\![\mu_2]\!]_\rho(t_2) + (1-p)[\![\mu_1]\!]_\rho(t_1).[\![\mu_2]\!]_\rho(t_2) \\
= \; & p.[\![\mu_0 + \mu_2]\!]_\rho(t_1 + t_2) + (1-p).[\![\mu_1 + \mu_2]\!]_\rho(t_1 + t_2) \\
= \; & [\![(\mu_0 + \mu_2) \oplus_p (\mu_1 + \mu_2)]\!]_\rho(t_1 + t_2)
\end{aligned}$$

The proofs for axioms (NP2) and (NP3) proceed in a similar way. The proofs for axioms (NP4) and (NP5) are straightforward.

To show completeness, let $\xi_0, \xi_1 \in \mathsf{T}(\Sigma)$ with $\xi_0 \sim \xi_1$. By Lemma 6.11 we can assume that ξ_0 and ξ_1 are in normal form. We proceed by induction on $n = \mathsf{depth}(\xi_0) + \mathit{depth}(\xi_1)$. If $\xi_0, \xi_1 \in \mathsf{T}(\Sigma)$, the proof proceeds like the non-probabilistic case [Mil89, Proposition 15]. If $\xi_0, \xi_1 \in \mathsf{DT}(\Sigma)$, then $\xi_0 = \bigoplus_{i \in I} p_i \delta(t_i)$ and $\xi_1 = \bigoplus_{j \in J} p_j \delta(t_j)$. Let $C = (\{t_i \mid i \in I\} \cup \{t_j \mid j \in J\})/\sim$ and define $\mathscr{C} = \bigcup_{c \in C} \{t_c\}$ with t_c a term in c. By inductive hypothesis, for all t_i with $i \in I$ there is $t_{i,c} \in \mathscr{C}$ s.t. $E_{CCS} \vdash t_i = t_{i,c}$. Similarly, for all $t_j, j \in J$, there is $t_{j,c} \in \mathscr{C}$ s.t. $E_{CCS} \vdash t_j = t_{j,c}$. Let ξ_0' (resp. ξ_1') obtained from ξ_0 (resp. ξ_1) by replacing each t_i by $t_{i,c}$ (resp. t_j by $t_{j,c}$). Using axioms (P1–P3), we can reorder and simplify both terms until they equate. □

In order to derive axioms for systems with rules including negative premises, we introduce the family of one-step restriction operators ∂_H^1, where $H \subseteq A$, $\mathsf{ar}(\partial_H^1) = s \to s$. The semantics of ∂_H^1 is given by

$$\frac{x \xrightarrow{a} \mu}{\partial_H^1(x) \xrightarrow{a} \mu} \quad a \notin H \tag{6.2}$$

$$\partial_H^1(x+y) = \partial_H^1(x) + \partial_H^1(y) \quad \text{(H1)} \qquad \partial_H^1(0) = 0 \quad \text{(H4)}$$
$$\partial_H^1(a.\mu) = a.\mu \quad \text{if } a \notin H \quad \text{(H2)} \qquad \partial_H^1(\mu_1 \oplus_p \mu_2) = \partial_H^1(\mu_1) \oplus_p \partial_H^1(\mu_2) \quad \text{(H5)}$$
$$\partial_H^1(a.\mu) = 0 \quad \text{if } a \in H \quad \text{(H3)} \qquad \partial_H^1(\delta(x)) = \delta(\partial_H^1(x)) \quad \text{(H6)}$$

Table 6.2: Axioms for ∂_H^1.

The term $\partial_H^1(t)$ cannot perform any action $a \in H$ in the next step, otherwise behaves as t. The signature Σ_{CCS^∂} extends Σ_{CCS} with operators ∂_H^1 for all $H \subseteq A$. The specification $P_{CCS^\partial} = (\Sigma_{CCS^\partial}, A, R_{CCS^\partial})$ is the PTSS whose set of rules R_{CCS^∂} extends R_{CCS} with the family of rules given in Equation 6.2.

Let E_{CCS^∂} be the set of equations E_{CCS} extended with the equations in Table 6.2. The equations H1–H4 are standard for the one-step restriction operator [ABV94]. Equations H5 and H6 propagate the one-step restriction operator to each single term in the support of a distribution. Hence, restriction distributes over probabilistic choices and Dirac embedding.

Theorem 6.14. *E_{CCS^∂} is sound and ground-complete for bisimilarity equivalence.*

Proof. Soundness of H1–H4 is proven just like in the non-probabilistic setting [ABV94]. Soundness of H5 and H6 is proven by showing that both sides of each axiom represent exactly the same distribution. ∂_H^1 can be eliminated in the sense that for each closed term $\xi \in T(\Sigma_{CCS^\partial})$ there is a closed term $\xi' \in T(\Sigma_{CCS})$ such that $E_{CCS^\partial} \vdash \xi = \xi'$. This can be proven by induction on the height of a term. (Notice that, when reading from left to right, axioms H1, H5, and H6 "push" operator ∂_H^1 inside the term, while axioms H2–H4, remove it.) Using elimination and Theorem 6.13, ground-completeness follows immediately. □

6.3.2 Probabilistically lifted operators

The semantics of all probabilistically lifted operators is defined following the same scheme. Thus, the axioms for these operators are defined similarly regardless if the original operator is distinctive, smooth or non-smooth. There are actually two types of axioms that explain how a lifted operator interacts with the probabilistic operations $\oplus_p$ and δ.

Definition 6.15. Let f be an operator with arity $\mathsf{ar}(f) = \sigma_1 \dots \sigma_{\mathsf{r}(f)} \to s$. We associate with f the axiom system E_f consisting of the following equations:

1. *Probabilistic distributivity laws:* For each position i of f, s.t. $\sigma_i = s$, and for each $p \in \mathbb{Q} \cap (0,1)$ we have the equations

$$f(\mu_1, .., \mu_i' \oplus_p \mu_i'', .., \mu_{\mathsf{r}(f)}) = f(\mu_1, .., \mu_i', .., \mu_{\mathsf{r}(f)}) \oplus_p f(\mu_1, .., \mu_i'', .., \mu_{\mathsf{r}(f)})$$

2. *Dirac distributivity laws:* We have the equation

$$f(\theta_1, \dots, \theta_{\mathsf{r}(f)}) = \delta(f(\zeta_1, \dots, \zeta_{\mathsf{r}(f)}))$$

with $\theta_i = \delta(\zeta_i), \zeta_i \in \mathcal{V}_s$ if $\sigma_i = s$ and $\theta_i = \zeta_i, \zeta_i \in \mathcal{V}_d$ if $\sigma_i = d$.

The soundness of these laws follows immediately from the semantics of $\oplus_p$, δ and the lifted operator, and using rational arithmetic. (Rational arithmetic be completely axiomatized for ground terms, which are the only ones we use, see e.g. [CMR97]).

6.3.3 Axiomatizing distinctive and smooth operators

A smooth rule is a rule that, whenever a variable is tested in a positive literal, then it is the only literal that tests that variable and it does not occur in the target of the conclusion. A smooth operator is an operator defined only by smooth rules. A distinctive operator is a smooth operator in which the hypothesis of the rules are mutually disjoint.

Definition 6.16. A generalized PGSOS rule is *smooth* if it has the form

$$\frac{\{x_i \xrightarrow{a_i} \mu_i \mid i \in I\} \quad \{x_j \xrightarrow{b_{j,n}} \not\rightarrow \mid j \in J, n \in N_j\}}{f(\zeta_1, \ldots, \zeta_{\mathsf{r}(f)}) \xrightarrow{a} \theta}$$

where I and J are disjoint sets s.t. $I \cup J = \{i \in \{1,..,\mathsf{r}(f)\} \mid \zeta_i \in \mathcal{V}_s\}$, and $x_i \notin \mathcal{V}(\theta)$ if $i \in I$. An operator f is *smooth* if all its defining rules are smooth.

A smooth operator f is *distinctive* if (i) each f-defining rule tests the same set of arguments I positively, and (ii) for every two different f-defining rules there is some argument $\zeta_i \in \mathcal{V}_s$ tested positively by both rules, but with a different action.

Notice that $+$ is smooth, but it is not distinctive since, e.g., x is tested positively in the first rule (actually, a set of rules), but not in the second one. Instead, ∂_H^1 is distinctive.

We introduce a new operator that we will use in our examples. Consider that each action a may fail with probability $p_a \in [0,1)$. In case of failure, the occurrence of a is ignored and the system remains in the same state, otherwise, it proceeds normally. The new operator $\mathsf{sc}(t)$ is a safe controller that minimizes the probability of failure of process t. Its semantics is given by the rules

$$\frac{x \xrightarrow{a} \mu \quad \{x \xrightarrow{b} \not\rightarrow \mid p_b < p_a\}}{\mathsf{sc}(x) \xrightarrow{a} \delta(x) \oplus_{p_a} \mathbf{sc}(\mu)} \qquad \frac{x \xrightarrow{a} \mu \quad p_a = 0}{\mathsf{sc}(x) \xrightarrow{a} \mathbf{sc}(\mu)}$$

sc is a variant of the ACP-style priority operator and it is not smooth since the rule on the left tests x in the positive literal but also in the negative literal, and, moreover, x appears in the target of the conclusion.

Let $pos(r) = I$ (resp. $neg(r) = J$) be those positions which are positively (resp. negatively) tested by rule r. Let $pact(r,i) = \{a_i \mid i \in I\}$ (resp. $nact(r,i) = \{b_{i,n} \mid n \in N_i\}$) be those actions for which x_i is positively (resp. negatively) tested by rule r. Note that if $N_i = \emptyset$ then $nact(r,i) = \emptyset$. A position i of operator f is *positive* if $i \in pos(r)$ for all rules r defining f.

Definition 6.17. Let f be a distinctive operator with arity $\mathsf{ar}(f) = \sigma_1 \ldots \sigma_{\mathsf{r}(f)} \to s$. Let $\zeta_i \in \mathcal{V}_s$ if $\sigma_i = s$ and $\zeta_i \in \mathcal{V}_d$ if $\sigma_i = d$ for $1 \le i \le \mathsf{r}(f)$. We associate with f the axiom system E_f consisting of the following equations:

1. *Nondeterministic distributivity laws:* For each positive position i of f, we have

$$f(\zeta_1, .., \zeta_i' + \zeta_i'', .., \zeta_{\mathsf{r}(f)}) = f(\zeta_1, .., \zeta_i', .., \zeta_{\mathsf{r}(f)}) + f(\zeta_1, .., \zeta_i'', .., \zeta_{\mathsf{r}(f)})$$

2. *Action laws:* For each f-defining rule r, we have the equation

$$\rho(f(\zeta_1,\ldots,\zeta_{\mathrm{r}(f)})) = a.\rho(\theta)$$

with $\mathrm{conc}(r) = f(\zeta_1,\ldots,\zeta_{\mathrm{r}(f)}) \xrightarrow{a} \theta$ and substitution ρ defined by

$$\rho(\zeta) = \begin{cases} a_i.\mu_i & \text{if } \zeta = x_i \text{ with } i \in pos(r) \\ \partial_H^1(x_i) & \text{if } \zeta = x_i \text{ with } i \in neg(r) \text{ and } H = nact(r,i) \neq \emptyset \\ \zeta & \text{otherwise.} \end{cases}$$

3. *Inaction laws:* We have the equations

$$\rho(f(\zeta_1,\ldots,\zeta_{\mathrm{r}(f)})) = 0$$

for all substitutions ρ mapping properly into terms of the form 0, x, $a.\mu$, $b.\mu + x$, or μ, such that for every f-defining rule r there is some position i with sort $\sigma_i = s$ satisfying one of the following conditions:

- if $i \in pos(r)$, then either $\rho(\zeta_i) = 0$ or $\rho(\zeta_i) = a.\mu_i$ with $a \notin pact(r,i)$, or
- if $i \in neg(r)$, then $\rho(\zeta_i) = b.\mu_i + x$ with $b \in nact(r,i)$.

The fact that all rules of a distinctive operator f test positively in the same positions guarantees the soundness of the nondeterministic distributivity law. There is one action law for each rule of f. The action law describe the execution of an action by pushing the executing action to the "head" of the term. The conditions of its associated rule are properly encode in each operand of f. Notice that the action laws can only be applied after f has been distributed over every sum. Contrarily to the action law, an inaction law traverse every f-defining rule ensuring through the operands that at least one of the conditions of each rule does not hold.

Soundness of axioms in E_f can be proven regardless of the PTSS containing operator f as long as the set of rules defining the semantics of f is the same for any PTSS. That is, if f is defined in a PTSS P, E_f is sound for any *disjoint extension* (Definition 2.41) of P. Then, we have the following theorem.

Theorem 6.18. *Let $P = (\Sigma, A, R)$ be a PTSS in generalized PGSOS format with $P_{CCS^\partial} \sqsubseteq P$ and $\Sigma_{sd} = \Sigma - \Sigma_{CCS^\partial}$ is a collection of distinctive operators. Let E_P be the axiom system consisting of E_{CCS^∂} and $E_f \cup E_f$, for each $f \in \Sigma_{sd}$. Then, for every disjoint extension $P' \sqsupseteq P$ in generalized PGSOS format, the axiom system E_P is sound for bisimilarity equivalence on P'.*

Proof. The soundness of the probabilistic distributivity laws and the Dirac distributivity laws (Definition 6.15) is straightforward following the reasoning used to prove that axiom (NP1) is sound (cf. proof of Theorem 6.13).

The soundness of the nondeterministic distributivity law and inaction law is straightforward by definition of distinctive operators and, for the latter, the conditions over ρ.

The soundness of action laws follows the reasoning used for the non-probabilistic case. Let $\rho(f(\zeta_1,\ldots,\zeta_{\mathrm{r}(f)})) = a.\rho(\theta)$ the axiom introduced by this law. Let ρ' be a closed

substitution. Given that all rules are distinctive, the term $\rho'(\rho(f(\zeta_1,\ldots,\zeta_{\mathsf{r}(f)})))$ can only execute the action a and then select the next state using distribution $[\![\theta]\!]_{\rho'}$ (recall that definition of distinctive rule ensures that terms $\rho(\zeta_1),\ldots,\rho(\zeta_{\mathsf{r}(f)})$ only can satisfy the premises of one rule of f). On the other hand, the term $\rho'(\rho(a.\theta))$ executes an action a and selects the next state using distribution $[\![\rho(\theta)]\!]_{\rho'}$. Because no variable ζ_i with $i \in I$ appears in θ, we have $[\![\theta]\!]_{\rho'} = [\![\rho(\theta)]\!]_{\rho'}$. Then both terms have exactly the same behavior, i.e. they are bisimilar. □

Notice that the set of rules R defining a smooth operator f in a PTSS P can always be partitioned into sets $R_1,\ldots,R_m$, such that f is distinctive when considering only the rules in R_i. Let f_i be fresh operators with arity $\mathsf{ar}(f_i) = \mathsf{ar}(f)$ and let R'_i be the same set of rules as R_i only that the operator in the source of each rule is renamed to f_i. Consider the disjoint extension $P' \sqsupseteq P$ with all fresh operators f_i and rules in $R'_1 \cup \ldots \cup R'_m$ added to the signature and set of rules of P, respectively. Then, it should be clear that the *distinctive law*

$$f(\zeta_1,\ldots,\zeta_{\mathsf{r}(f)}) = f_1(\zeta_1,\ldots,\zeta_{\mathsf{r}(f)}) + \cdots + f_m(\zeta_1,\ldots,\zeta_{\mathsf{r}(f)}) \tag{6.3}$$

is sound for bisimulation. Thus, a smooth operator f is axiomatized by the nondeterministic choice over the distinctive variants of f.

Theorem 6.19. *Let $P = (\Sigma, A, R)$ be a PTSS in PGSOS format with $P_{CCS^\partial} \sqsubseteq P$ and let $f \in \Sigma$ be a smooth operator. There is a disjoint extension $P' = (\Sigma', A, R')$ of P with m distinctive smooth operators $f_1,\ldots,f_m$ s.t. $\mathsf{ar}(f_i) = \mathsf{ar}(f)$ for $1 \le i \le m$ and Equation 6.3 is sound for bisimilarity equivalence in any disjoint extension of P'.*

Proof. Let $P' \sqsupseteq P$ be a PTSS in PGSOS format with distinctive operators $f_1,\ldots,f_m$ s.t. $\mathsf{ar}(f_i) = \mathsf{ar}(f)$ with $i \in \{1,\ldots,m\}$ and the transition rules for each f_i, after renaming f_i by f, form a partitioning of all the rules of f in P. Intuitively, the operators $f_1,\ldots,f_{\mathsf{r}(f)}$ are the distinctive version of f. Now the transition $f(\xi_1,\ldots,\xi_{\mathsf{r}(f)}) \xrightarrow{a} \pi$ iff $f_r(\xi_1,\ldots,\xi_{\mathsf{r}(f)}) \xrightarrow{a} \pi$ for some $r \in \{1,\ldots,m\}$ and all $\xi_1,\ldots,\xi_{\mathsf{r}(f)} \in \mathsf{T}(\Sigma)$. Therefore we get that $f(\xi_1,\ldots,\xi_{\mathsf{r}(f)})$ and $f_1(\xi_1,\ldots,\xi_{\mathsf{r}(f)}) + \ldots + f_{n_f}(\xi_1,\ldots,\xi_{\mathsf{r}(f)})$ are bisimilar for all all $\xi_1,\ldots,\xi_{\mathsf{r}(f)} \in \mathsf{T}(\Sigma)$. □

6.3.4 Axiomatizing non-smooth operators

An operator that is not smooth has a rule in which a variable that is tested in a positive literal either is tested in a second literal or it appears in the target of a conclusion. In this case we proceed by constructing a smooth version of the operator with one argument for each kind of use of the variable that breaks smoothness (actually, one argument for each positive test plus an additional one if the variable is tested negatively or it appears on the target of the conclusion of a rule). Thus, for the unary operator sc, we introduce a binary operator $\overline{\mathsf{sc}}$, the first argument related to the positive literal and the other related to the negative test and the occurrence in the target of the rule. So $\overline{\mathsf{sc}}$ is defined by the rules

$$\frac{x \xrightarrow{a} \mu \quad \{y \overset{b}{\not\to} \mid p_b < p_a\}}{\overline{\mathsf{sc}}(x,y) \xrightarrow{a} \delta(y) \oplus_{p_a} \mathbf{sc}(\mu)}, \text{ if } p_a > 0 \qquad \frac{x \xrightarrow{a} \mu}{\overline{\mathsf{sc}}(x,y) \xrightarrow{a} \mathbf{sc}(\mu)}, \text{ if } p_a = 0$$

$$\overline{\mathsf{sc}}(x_1 + x_2, y) = \overline{\mathsf{sc}}(x_1, y) + \overline{\mathsf{sc}}(x_2, y) \qquad \overline{\mathsf{sc}}(a.\mu, y) = a.\mathbf{sc}(\mu) \quad \text{if } p_a = 0$$

$$\overline{\mathsf{sc}}(a.\mu, \partial_H^1(x)) = a.(\delta(\partial_H^1(x)) \oplus_{p_a} \mathbf{sc}(\mu)) \qquad \overline{\mathsf{sc}}(0, y) = 0$$

$$\text{with } H = \{b \mid p_b < p_a\} \qquad \overline{\mathsf{sc}}(a.\mu, b.\nu + y) = 0 \quad \text{if } p_b < p_a$$

$$\overline{\mathbf{sc}}(\mu_1 \oplus_p \mu_2, \nu) = \overline{\mathbf{sc}}(\mu_1, \nu) \oplus_p \overline{\mathbf{sc}}(\mu_2, \nu) \qquad \overline{\mathbf{sc}}(\delta(x), \delta(y)) = \delta(\overline{\mathsf{sc}}(x, y))$$

$$\overline{\mathbf{sc}}(\mu, \nu_1 \oplus_p \nu_2) = \overline{\mathbf{sc}}(\mu, \nu_1) \oplus_p \overline{\mathbf{sc}}(\mu, \nu_2)$$

Table 6.3: Axiomatization of sc (redundant laws, such as $\overline{\mathsf{sc}}(0, a.\mu) = 0$, are omitted)

It should be clear that $\mathsf{sc}(x) = \overline{\mathsf{sc}}(x, x)$. Moreover, notice that $\overline{\mathsf{sc}}$ is smooth. (In fact, it is also distinctive.) The premise on the second rule could have alternatively tested on y rather than x, in which case, $\overline{\mathsf{sc}}$ would have also been smooth but not distinctive.

In general, given a non-smooth operator f, we define a new smooth operator f' by extending its arity as explained above, and proceeding as following: for each rule r of f we introduce a new rule r' for f' such that, if we intend to equate $f(\vec{\zeta}) = f'(\vec{\zeta'})$, and $f(\vec{\varsigma})$ and $f'(\vec{\varsigma'})$ are the sources of r and r', respectively, $r[\vec{\zeta}/\vec{\varsigma}]$ and $r'[\vec{\zeta'}/\vec{\varsigma'}]$ have to be identical with the exception of their sources. (Here, $[\vec{\zeta}/\vec{\varsigma}]$ denotes the usual substitution of variables.) Notice that this results in a one to one correspondence between the rules of f and those of f'. Then, we have the following theorem.

Theorem 6.20. *Let $P = (\Sigma, A, R)$ be a PTSS in generalized PGSOS format with $P_{CCS^\partial} \sqsubseteq P$. Let $f \in \Sigma_P$ be a non-smooth operator. Then there is a disjoint extension $P' \sqsupseteq P$ with a smooth operator f' s.t. the equation $f(\zeta_1, \ldots, \zeta_{\mathsf{r}(f)}) = f'(\zeta'_1, \ldots, \zeta'_{\mathsf{r}(f')})$, where ζ_i, $1 \leq i \leq \mathsf{r}(f)$ are all different variables and $\{\zeta'_1, \ldots, \zeta'_{\mathsf{r}(f')}\} \subseteq \{\zeta_1, \ldots, \zeta_{\mathsf{r}(f)}\}$, is sound for bisimilarity equivalence in every disjoint extension of P'.*

Proof. This theorem can be shown by applying the same argumentation as in the proof of Lemma 4.13 of [ABV94]: The *barb-factor* is used to determine the number of copies needed by the operator f'. Then, adapting Lemma 4.12 of [ABV94] allows to show that for all closed substitution ρ, $\rho(f(\zeta_1, \ldots, \zeta_{\mathsf{r}(f)})) \xrightarrow{a} \pi$ iff $\rho(f'(\zeta'_1, \ldots, \zeta'_{\mathsf{r}(f')})) \xrightarrow{a} \pi$. □

As an example, we complete the axiomatization of sc with the axioms for $\overline{\mathsf{sc}}$ which can be derived using Definitions 6.15 and 6.16. They are given in Table 6.3.

As a result of the previous theorems we obtain the algorithm of Figure 6.1. Given a PTSS $P_i = (\Sigma^i, A, R^i)$ in generalized PGSOS format, the algorithm generates an equational theory E_o that captures the behavior of all operators in P_i and is sound for bisimilarity equivalence.

The fact that the set of rules of P_i (and hence also P_o) is finite guarantees that the equational theory E_o is *head-normalizing* for all operators of $P_o = (\Sigma^o, A, R^o)$, that is, every closed term of P_o can be proven equal to a term of the form 0, $\sum_{i \in I} a_i.\theta_i$ or $\bigoplus_{j \in J} p_j \delta(t_j)$, with $\theta_i \in T(\Sigma^o_d)$ and $t_j \in T(\Sigma^o_s)$, within the equational theory E_o. The construction of head-normal forms is the key towards proving ground-completeness. In fact, notice that if the semantics of a term $t \in T(\Sigma^o_s)$ is a finite tree, then all operators can be eliminated in E_o (i.e., there is a term $t' \in T(\Sigma_{CCS})$, s.t., $E_o \vdash t = t'$). However this is not the case

Input: a PTSS P_i in generalized PGSOS format

Output: a PTSS P_o in generalized PGSOS format, with $P_o \supseteq P_i$, and an equational theory E_o that is sound for bisimilarity equivalence in all disjoint extensions of P_o.

1. If necessary, complete P_i so that it disjointly extends P_{CCS^∂}.
2. For each non-smooth operator of P_i, extend the system with a smooth version according to Theorem 6.20 and add all the corresponding equations.
3. For each smooth non-distinctive operator $f \notin \Sigma_{CCS^\partial}$ in the resulting PTSS, apply the construction of Theorem 6.19 and extend the PTSS with the distinctive operators $f_1, \ldots, f_m$ and the respective rules. Add also the resulting instances of axiom (6.3).
4. Add all equations associated to the distinctive operators in the resulting system (but not in Σ_{CCS^∂}) according to Definition 6.17.
5. Finally, for every operator not in Σ_{CCS^∂} add the equation for their respective lifted version according to Definition 6.15.

Figure 6.1: Algorithm to generate an axiomatization for P_i

in general. Consider the constant operator nwf whose semantics is defined by the rule $\mathsf{nwf} \xrightarrow{a} \delta(\mathsf{nwf}) \oplus_{\frac{1}{2}} \delta(0)$. Using the action law, axiom $\mathsf{nwf} = a.(\delta(\mathsf{nwf}) \oplus_{\frac{1}{2}} \delta(0))$ is derived, in which the elimination process will never terminate.

In order to guarantee ground-completeness, we adapt the notion of semantically well-founded of [ABV94] to our setting. Given a PTSS $P = (\Sigma, A, R)$, a term $t \in \mathsf{T}(\Sigma)$ is *semantically well-founded* in P if there is no infinite sequence $t_0\, a_0\, \theta_0\, t_1\, a_1\, \theta_1 \ldots$ of terms $t_i \in \mathsf{T}(\Sigma)$ and $\theta_i \in \mathsf{DT}(\Sigma)$ and actions $a_i \in A$, such that $t_i \xrightarrow{a_i} \theta_i$ is derivable in P and $[\![\theta_i]\!](t_{i+1}) > 0$, for all $i \geq 0$. P is *semantically well-founded* if all its terms are. Now, if P_o is semantically well-founded (which is the case if P_i is semantically well-founded and $P_{CCS} \sqsubseteq P_i$), E_o has an elimination theorem. Before we can show that the equational theory E_o is ground complete for bisimilarity equivalence (Theorem 6.24), we need to prove that E_o is head normalizing (Lemma 6.23). First we need to show that E_{CCS^∂} allows to derive equality between terms when restricting on actions that the terms cannot perform.

Lemma 6.21 ([ABV94])**.** *Let $t = \sum_{i \in I} a_i.\theta_i \in T(\Sigma)$ be a closed state term and $B \subseteq A$ any set of actions with $a_i \notin B$ for all $i \in I$. Then $E_{CCS^\partial} \vdash t = \partial_B^1(t)$.*

Next we need to show that E_o allows to normalize terms that are constructed by applying operators to terms in normal form.

Lemma 6.22. *Let $f \in \Sigma$ be a function and $\xi_1, \ldots, \xi_{\mathsf{r}(f)} \in T(\Sigma)$, $\theta_1, \ldots, \theta_{\mathsf{r}(f)} \in DT(\Sigma)$ be terms in head normal form. Then there are $t \in T(\Sigma)$ and $\theta \in DT(\Sigma)$ s.t. t and θ are in normal form, and $E_o \vdash f(\xi_1, \ldots, \xi_{\mathsf{r}(f)}) = t$ and $E_o \vdash \boldsymbol{f}(\theta_1, \ldots, \theta_{\mathsf{r}(f)}) = \theta$.*

Proof. By induction over the term structure. A trigger of a rule defines which actions the argument processes need to be able to perform or should not be able to perform s.t. the

rule is applicable. Formally, a trigger of a smooth rule r is a tuple $\langle e_1, \ldots, e_{\mathsf{r}(f)} \rangle$ with

$$e_i = \begin{cases} a_i & \text{if } i \in I \\ \{b_{i,n} \mid n \in N_i\} & \text{if } i \in J \\ \emptyset & \text{otherwise (i.e. } \sigma_i = d) \end{cases}$$

Let $\mathsf{ar}(f) = \sigma_1 \ldots \sigma_{\mathsf{r}(f)} \to s$. We start by the state operator $f \in \Sigma_s$.

Case 1. There is an argument i that is tested positively by f and for which ξ_i has the form $t_i' + t_i''$ with t_i', t_i'' are in head normal form. Applying one of the axioms introduced by the nondeterministic distributivity laws we get

$$E_o \vdash f(\xi_1, \ldots, t_i' + t_i'', \ldots, \xi_{\mathsf{r}(f)}) = f(\xi_1, \ldots, t_i', \ldots \xi_{\mathsf{r}(f)}) + f(\xi_1, \ldots, t_i'', \ldots \xi_{\mathsf{r}(f)})$$

By the induction hypothesis, there are $t', t'' \in \mathsf{T}(\Sigma)$ in head normal form such that $E_o \vdash f(\xi_1, \ldots, t_i', \ldots \xi_{\mathsf{r}(f)}) = t'$ and $E_o \vdash f(\xi_1, \ldots, t_i'', \ldots \xi_{\mathsf{r}(f)}) = t''$. Then $E_o \vdash f(\xi_1, \ldots, t_i' + t_i'', \ldots, \xi_{\mathsf{r}(f)}) = t' + t''$. and $t' + t''$ is in head normal form.

Case 2. There is an argument i that is tested positively by f and for which $\xi_i = 0$. Since f is distinctive, all rules for f test i positively. Thus E_o contains a deadlock law $f(\zeta_1, \ldots, \zeta_{i-1}, 0, \zeta_{i+1}, \ldots, \zeta_{\mathsf{r}(f)}) = 0$. Instantiation of this axiom gives $E_o \vdash f(t_1, \ldots, t_{\mathsf{r}(f)}) = 0$ and 0 is in head normal form.

Case 3. For all arguments k that are tested positively by f, ξ_k has the form $a_k.\pi_k$. Two sub-cases arise.

Case 3.1. For each rule for f with trigger $\langle e_1, \ldots, e_{\mathsf{r}(f)} \rangle$, there is an i that is tested positively s.t. $e_i \neq a_i$. Then E_o contains a deadlock law $f(\hat{\xi}_1, \ldots, \hat{\xi}_{\mathsf{r}(k)}) = 0$ where $\hat{\xi}_k = a_k.\mu_k$ if k is tested positively, and $\hat{t}_k = x_k$ otherwise. Instantiation of this law gives $E_o \vdash f(\xi_1, \ldots, \xi_{\mathsf{r}(f)}) = 0$.

Case 3.2. There exists a rule r for f with trigger $\langle e_1, \ldots, e_{\mathsf{r}(f)} \rangle$ s.t. $e_k = a_k$ for all k that are tested positively. Since f is distinctive, r is the unique rule with this property. Here, two new sub-cases arise.

Case 3.2.1. There is an index j that is not tested positively, and there is an action $b \in e_j$ s.t. $E_o \vdash \xi_j = b.t_j' + t_j''$. Notice E_o contains a deadlock law $f(\hat{\xi}_1, \ldots, \hat{\xi}_{\mathsf{r}(k)}) = 0$ s.t. $\hat{\xi}_k = a_k.\mu_k$ if k is tested positively, $\hat{\xi}_k = b.\theta_k' + t_k''$ if $k = j$ and $\hat{\xi}_k = x_k$ otherwise. Then $E_o \vdash f(\xi_1, \ldots, b.t_k' + t_k'', \ldots, \xi_{\mathsf{r}(f)}) = 0$.

Case 3.2.2. For each index n that is not tested positively and $\sigma_n = s$, ξ_n has the form $\sum a_{i,j} P_{i,j}$ with $a_{i,j} \notin e_n$ for all j. Applying Lemma 6.21, for all n, s.t. $e_n \neq \emptyset$ and $e_n \subset A$, term ξ_n can be replaced by $\partial^1_{e_n}(\xi_n)$. Applying the action law corresponding to r is enough to get the required normal form.

We proceed by the distribution operator $\boldsymbol{f} \in \Sigma_d$.

Case 4. Suppose that all arguments of $\boldsymbol{f}$ have the form $\theta_i = \delta(\tau_i)$ for $\sigma_i = s$ and some $\tau_i \in \mathsf{T}(\Sigma)$, otherwise $\theta_i = \tau_i \in \mathsf{DT}(\Sigma)$. Applying the Dirac distributivity law, $E_o \vdash \boldsymbol{f}(\theta_i, \ldots, \theta_{\mathsf{r}(f)}) = \delta(f(\tau_1, \ldots, \tau_{\mathsf{r}(f)}))$ and the right side is a term in head normal form.

Case 5. Let θ_j be an argument of $\boldsymbol{f}$ s.t. $\sigma_j = s$ and θ_j does not have the form $\delta(t_j)$. Then, $\theta_j = \theta_j' \oplus_p \theta_j''$ with θ_j', θ_j'' in head normal form. Applying the corresponding probabilistic distributivity laws gives $E_o \vdash \boldsymbol{f}(\theta_1, \ldots, \theta_j' \oplus_p \theta_j'', \ldots, \theta_{\mathsf{r}(f)}) = \boldsymbol{f}(\theta_1, \ldots, \theta_j', \ldots, \theta_{\mathsf{r}(f)}) \oplus_p \boldsymbol{f}(\theta_1, \ldots, \theta_j'', \ldots, \theta_{\mathsf{r}(f)})$. By the induction hypothesis, there are $\theta', \theta'' \in \mathsf{DT}(\Sigma)$ in head

normal form s.t. $E_o \vdash \boldsymbol{f}(\theta_1,\ldots,\theta'_j,\ldots,\theta_{\mathrm{r}(f)}) = \theta'$ and $E_o \vdash \boldsymbol{f}(\theta_1,\ldots,\theta''_j,\ldots,\theta_{\mathrm{r}(f)}) = \theta''$. Then $E_0 \vdash \boldsymbol{f}(\theta_1,\ldots,\theta'_j \oplus_p \theta''_j,\ldots,\theta_{\mathrm{r}(f)}) = \theta' \oplus_p \theta''$ and $\theta' \oplus_p \theta''$ is in head normal form. □

Lemma 6.23. *Let $P_o = (\Sigma, A, R)$ be the PTSS in generalized PGSOS format and E_o the equational theory of Figure 6.1. Then E_o is head normalizing for all operators of P_o.*

Proof. The proof follows closely the argumentation of the non-probabilistic setting [ABV94]. Notice that the axioms added in the steps 2 and 3 of the algorithm (see Figure 6.1) allows us to focus only on the case where the operator is distinctive. The thesis follows via structural induction using Lemma 6.22. □

Now we can show that the algorithm of Figure 6.1 generates an equational theory E_o that is ground-complete for bisimilarity equivalence in P_o.

Theorem 6.24. *Let P_i be the input and P_o and E_o be the outputs of the algorithm in Figure 6.1. If P_o is a semantically well-founded PTSS, then the equational theory E_o is ground-complete for bisimilarity equivalence in P_o.*

Proof. The argumentation follows closely the proof of Theorem 5.2 in [ABV94]. Since P_o is finitely branching, we can define the function $maxt(t)$ that gives for each well-founded term t the maximal number of consecutive transitions that t can execute.

Let t and u be two semantically well-founded state terms with $t \sim u$. By induction on $maxt(t)$, we show that t is provably equal to a term in $T(\Sigma_{CCS})$. Since E_o is head normalizing for P_o, there is a term t' with form $\sum a_i t_i$ s.t. $E_o \vdash t = t'$. It is easy to show that $\sim$ preserves the maximum number of consecutive transitions, then $maxt(t) = maxt(t')$. In addition, if $t \xrightarrow{a_i} t_i$ then $maxt(t_i) < maxt(t)$ for all i. By the induction hypothesis, for each i, there exists $t'_i \in T(\Sigma_{CCS})$ s.t. $E_o \vdash t_i = t'_i$. The induction step follows because $E_o \vdash t = t''$ with $t'' = \sum a_i t'_i$.

In the same way we can prove that there is $u'' \in T(\Sigma_{CCS})$ s.t. $E_o \vdash u = u''$. Finally, completeness follows by Theorem 6.13. □

Ground completeness can be extended to semantically non well-founded PTSS in generalized PGSOS format by using the approximation induction principle (AIP) [BBK87]. We omit it here. The proof follows closely the lines of [ABV94].

6.4 Axiomatization of the bisimilarity metric

In the previous section we developed an equational theory for bisimilarity equivalence. Now we shift our focus to bisimilarity pseudometrics and develop an equational theory that characterizes the bisimulation distance.

6.4.1 Axiomatizing finite probabilistic trees

Let E^m_{CCS} be the system of equations in Table 6.4. The equations consider two kind of symbols for metrics: one on state terms (d) and the other on distribution terms ($\mathfrak{d}$). Axioms D1–D4 correspond to conditions (i) and (ii) of the definition of a pseudometric. Axioms

$$\mathsf{d}(x,x)=0 \quad \text{(D1)} \qquad \mathfrak{d}(\mu,\mu)=0 \quad \text{(D3)}$$

$$\mathsf{d}(x,y)=\mathsf{d}(y,x) \quad \text{(D2)} \qquad \mathfrak{d}(\mu,\nu)=\mathfrak{d}(\nu,\mu) \quad \text{(D4)}$$

$$\mathsf{d}(t,x)=\mathsf{d}(t',x) \quad \text{where } t=t' \text{ is one of axioms N1–N4} \quad \text{(MN)}$$

$$\mathfrak{d}(\theta,\mu)=\mathfrak{d}(\theta',\mu) \quad \text{where } \theta=\theta' \text{ is one of axioms NP1–NP5 or P1–P3} \quad \text{(MP)}$$

$$\mathsf{d}(0,a.\mu+x)=1 \quad \text{(H1)}$$

$$\mathsf{d}\Big(\textstyle\sum_{i\in I} a_i.\mu_i, \sum_{j\in J} b_j.\nu_j\Big) = \max\Big\{ \max_{i\in I} \min_{j\in J, a_i=b_j} \lambda\cdot\mathfrak{d}(\mu_i,\nu_j), \max_{j\in J} \min_{i\in I, a_i=b_j} \lambda\cdot\mathfrak{d}(\mu_i,\nu_j)\Big\} \quad \text{(H2)}$$

$$\mathfrak{d}\Big(\bigoplus_{i\in I} p_i\delta(x_i), \bigoplus_{j\in J} q_j\delta(y_j)\Big) = \min_{\omega\in\Omega(I,J)} \textstyle\sum_{i\in I, j\in J} \mathsf{d}(x_i,y_j)\cdot\omega(i,j) \quad \text{(K)}$$

where $\Omega(I,J)=\{\omega : I\times J\to[0,1] \mid \forall i\in I : \omega(i,J)=p_i, \forall j\in J : \omega(I,j)=q_j\}$

Table 6.4: Axiomatization of bisimilarity metric of CCS. (We assume $\min\varnothing=1$.)

MN and MP lift the axioms for bisimulation equivalence to bisimulation metrics. In a way, they state that two bisimilar terms should have the same distance to a third term. Axioms H1 and H2 correspond to the definition of the Hausdorff pseudometric (Definition 2.19). Finally, axiom K corresponds to the definition of the Kantorovich pseudometric (Definition 2.17). We also need the following general rules that should be consider together with the usual rules for reasoning on equational logic. For all $f : \sigma_1 \dots \sigma_{\mathsf{r}(f)} \to s$ and $g : \sigma_1 \dots \sigma_{\mathsf{r}(g)} \to d$, we have

$$\frac{\{\mathsf{d}(\zeta_i,\zeta_i')=0, \mathfrak{d}(\zeta_j,\zeta_j')=0 \mid 1\le i,j\le \mathsf{r}(f), \sigma_i=s, \sigma_j=d\}}{\mathsf{d}(f(\zeta_1,..,\zeta_{\mathsf{r}(f)}),z)=\mathsf{d}(f(\zeta_1',..,\zeta_{\mathsf{r}(f)}'),z)} \quad \text{(S1)}$$

$$\frac{\{\mathsf{d}(\zeta_i,\zeta_i')=0, \mathfrak{d}(\zeta_j,\zeta_j')=0 \mid 1\le i,j\le \mathsf{r}(f), \sigma_i=s, \sigma_j=d\}}{\mathfrak{d}(g(\zeta_1,..,\zeta_{\mathsf{r}(g)}),z)=\mathfrak{d}(g(\zeta_1',..,\zeta_{\mathsf{r}(g)}'),z)} \quad \text{(S2)}$$

These rules ensure that $E^m_{CCS} \vdash \mathsf{d}(t,t'')=\mathsf{d}(t',t'')$ whenever $E_{CCS} \vdash t=t'$ and similarly for distribution terms.

Let $\mathbf{d}$ be the bisimilarity metric and $\mathbf{K}(\mathbf{d})$ its Kantorovich lifting. Let ρ be a closed substitution. We define $[\![\mathsf{d}(t,t')]\!]_\rho = \mathbf{d}(\rho(t),\rho(t'))$ and $[\![\mathfrak{d}(\theta,\theta')]\!]_\rho = \mathbf{K}(\mathbf{d})([\![\rho(\theta)]\!],[\![\rho(\theta')]\!])$ for $t,t'\in\mathbb{T}(\Sigma)$ and $\theta,\theta'\in\mathbb{DT}(\Sigma)$. We lift $[\![\_]\!]_\rho$ to arithmetic terms containing expressions of the form $\mathsf{d}(t,t')$ or $\mathfrak{d}(\theta,\theta')$ in the obvious way, for instance $[\![\min_{i\in I} expr_i]\!]_\rho = \min_{i\in I}[\![expr_i]\!]_\rho$. E^m_{CCS} is sound for $\mathbf{d}$ in the sense that, whenever $E^m_{CCS} \vdash expr = expr'$ (meaning that $expr = expr'$ can be proved using axioms in E^m_{CCS} and arithmetic), then $[\![expr]\!]_\rho = [\![expr']\!]_\rho$ for every closed substitution ρ. Soundness should be clear for all the axioms except maybe for H2. By definition of bisimulation metric, the right-hand side is

smaller than or equal to the left-hand side interpreting them on any closed substitution. Equality follows from the fact that $\mathbf{d}$ is the *smallest* bisimulation metric.

Besides, E^m_{CCS} is also ground-complete for $\mathbf{d}$, in the sense that, for any (closed) arithmetic expressions *expr* and *expr*$'$ possibly containing closed terms of the form $\mathsf{d}(t,t')$ or $\mathfrak{d}(\theta,\theta')$ with $t,t' \in \mathsf{T}(\Sigma)$ and $\theta,\theta' \in \mathsf{DT}(\Sigma)$, $[\![expr]\!] = [\![expr']\!]$ implies $E^m_{CCS} \vdash expr = expr'$. Notice that by arithmetic, this is a direct consequence of the following claims: (i) for all closed state terms $t,t' \in \mathsf{T}(\Sigma)$ and $p \in [0,1]$, if $\mathbf{d}(t,t') = p$ then $E^m_{CCS} \vdash \mathsf{d}(t,t') = p$, and (ii) for all closed distribution terms $\theta,\theta' \in \mathsf{DT}(\Sigma)$, if $\mathbf{K}(\mathbf{d})([\![\theta]\!],[\![\theta']\!]) = p$, $E^m_{CCS} \vdash \mathfrak{d}(\theta,\theta') = p$. The proof of these claims follows by reducing closed terms involved in $\mathsf{d}(t,t')$ and $\mathfrak{d}(\theta,\theta')$ to normal form using axioms D1–D4, MN, and MP (and rules S1 and S2), and then inductively applying H1, H2, K and arithmetic calculations to reach the expected value. The following Theorems 6.25 and 6.28 are the metric variant of Theorem 6.13.

Theorem 6.25. *E^m_{CCS} is sound for the bisimilarity metric $\mathbf{d}$.*

Proof. Let ρ be any closed substitution. Because $[\![\mathsf{d}(\cdot,\cdot)]\!]_\rho$ is by definition a pseudometric on $\mathsf{T}(\Sigma)$ and $[\![\mathfrak{d}(\cdot,\cdot)]\!]_\rho$ is by definition a pseudometric on $\Delta(\mathsf{T}(\Sigma))$ the axioms D1,D2 and resp. axioms D3,D4 clearly hold. Axioms MN follows directly from the fact that bisimilarity equivalence is the kernel of $\mathbf{d}$ i.e. soundness of axioms N1–N4 w.r.t. bisimilarity equivalence (Theorem 6.13) gives soundness of MN w.r.t. the bisimilarity metric $\mathbf{d}$. As an example, take axiom N1. By soundness of E_{CCS}, $\rho(x)+\rho(y) \sim \rho(y)+\rho(x)$ for any closed substitution ρ, and hence $\mathbf{d}(\rho(x)+\rho(y),\rho(y)+\rho(x)) = 0$. Then

$$\begin{aligned}
& \mathbf{d}(\rho(x)+\rho(y),\rho(z)) \\
\leq\ & \mathbf{d}(\rho(x)+\rho(y),\rho(y)+\rho(x)) + \mathbf{d}(\rho(y)+\rho(x),\rho(z)) \\
=\ & \mathbf{d}(\rho(y)+\rho(x),\rho(z)) \\
\leq\ & \mathbf{d}(\rho(y)+\rho(x),\rho(x)+\rho(y)) + \mathbf{d}(\rho(x)+\rho(y),\rho(z)) \\
=\ & \mathbf{d}(\rho(x)+\rho(y),\rho(z))
\end{aligned}$$

which proves soundness of MN for this case. Similar calculations yield soundness of MP using $\mathbf{K}(\mathbf{d})$ instead of $\mathbf{d}$. Axiom H1 follows directly from the definition of bisimulation metric. In detail, if $\mathbf{d}(0,\rho(a.\mu+x))$ would be strictly less than 1, then the transition $\rho(a.\mu+x) \xrightarrow{a} \rho(\mu)$ would need to be matched by some a-labeled transition of 0. As the the inactive process 0 cannot perform any action, we necessarily have $\mathbf{d}(0,\rho(a.\mu+x)) = 1$. For axiom H2, we calculate

$$\begin{aligned}
& \mathbf{d}(\textstyle\sum_{i\in I} a_i.\rho(\mu_i), \sum_{j\in J} b_j.\rho(\nu_j)) \\
=\ & \max_{a\in A} \mathbf{H}(\lambda\cdot\mathbf{K}(\mathbf{d}))(\{\rho(\mu_i) \mid i\in I, a=a_i\},\{\rho(\nu_j) \mid j\in J, a=b_j\}) \\
=\ & \max_{a\in A} \max\{\max_{i\in I,a=a_i} \min_{j\in J,a=b_j} \lambda\cdot\mathbf{K}(\mathbf{d})(\rho(\mu_i),\rho(\nu_j)), \max_{j\in J,a=b_j} \min_{i\in I,a=a_i} \lambda\cdot\mathbf{K}(\mathbf{d})(\rho(\mu_i),\rho(\nu_j))\} \\
=\ & \max\{\max_{i\in I} \min_{j\in J,a_i=b_j} \lambda\cdot\mathbf{K}(\mathbf{d})(\rho(\mu_i),\rho(\nu_j)), \max_{j\in J} \min_{i\in I,a_i=b_j} \lambda\cdot\mathbf{K}(\mathbf{d})(\rho(\mu_i),\rho(\nu_j))\}
\end{aligned}$$

The first equality is a consequence that $\mathbf{d}$ is the smallest bisimulation metric. (Because $\mathbf{d}$ is a bisimulation metric we have that the first term has a value larger than or equal to the

second one; because it is the smallest one, equality follows.) The second equality is the definition of Hausdorff pseudometric, and the last equality is just arithmetic manipulation. Finally, soundness of axiom K follows immediately by the definition of Kantorovich pseudometric. □

To prove ground-completeness, we first need to reduce closed terms in expressions of the form $\mathsf{d}(t,t')$ and $\mathfrak{d}(\theta,\theta')$ to their normal form. We actually prove a more general property from which existence of normal form follows.

Lemma 6.26. *The following two statements hold:*

(a) $E_{CCS} \vdash t = t'$ *implies* $E^m_{CCS} \vdash d(t,z) = d(t',z)$ *for all* $t,t' \in \mathbb{T}(\Sigma)$ *and* $z \in \mathcal{V}_s$.

(b) $E_{CCS} \vdash \theta = \theta'$ *implies* $E^m_{CCS} \vdash \mathfrak{d}(\theta,\mu) = \mathfrak{d}(\theta',\mu)$ *for all* $\theta,\theta' \in \mathbb{DT}(\Sigma)$ *and* $\mu \in \mathcal{V}_d$.

Proof. We first recall that a proof of an equality e in a theory E is a list of equalities P such that $P[\#P] = e$ and for all $i \in \{1 \dots \#P\}$, either (i) $P[i]$ is an axiom, or (ii) $P[i]$ is the conclusion of a rule r and for every premise p of r there is a $j < i$ s.t. $P[j] = p$. An axiom can be any included on the theory E or the reflexivity axiom of equational reasoning and a rule could be any rule included in E or the rules for equational reasoning. (Actually an axiom is a rule with no premises, but we prefer to make the distinction here.)

By induction on the length of the proof P of $E_{CCS} \vdash \xi = \xi'$, we construct a proof P' in E^m_{CCS} such that $P'[\#P'] = \Big(\mathsf{d}(\xi,z) = \mathsf{d}(\xi',z)\Big)$ with a fresh $z \in \mathcal{V}_s$, if $\xi,\xi' \in \mathbb{T}(\Sigma)$, or $P'[\#P'] = \Big(\mathfrak{d}(\xi,\mu) = \mathfrak{d}(\xi',\mu)\Big)$ with a fresh $\mu \in \mathcal{V}_d$, if $\xi,\xi' \in \mathbb{DT}(\Sigma)$.

The base case can only result by applying axioms. This is already considered in the inductive case so we only focus on it. By induction we assume that all positions $j < i$ of P have already been encoded in a proof P' of length k_{i-1} and that for each $P[j] = (\xi = \xi')$ there is a position $k_j \leq k_{i-1}$ such that $P'[k_j] = \Big(\mathsf{d}(\xi,z) = \mathsf{d}(\xi',z)\Big)$ with fresh $z \in \mathcal{V}_s$ (or $P'[k_j] = \Big(\mathfrak{d}(\xi,\mu) = \mathfrak{d}(\xi',\mu)\Big)$ with fresh $\mu \in \mathcal{V}_d$, depending of the sort of ξ and ξ') has been proven with the k_j prefix of P'.

We now proceed by case analysis. We will focus only on state terms. The cases of distribution terms follow in the same way.

Suppose $P[i] = (t = t') \in E_{CCS}$ with $t,t' \in \mathbb{T}(\Sigma)$. Since it is an axiom in E_{CCS}, we use MN and let $P'[k_i] = \Big(\mathsf{d}(t,z) = \mathsf{d}(t',z)\Big)$, with fresh z, defining $k_i = k_{i-1} + 1$.

The case in which $P[i] = (t = t)$ was calculated using reflexivity proceeds in the same way.

Suppose $P[i] = (t = t')$ was proven using transitivity. Then there are $j,j' < i$ and $t'' \in \mathbb{T}(\Sigma)$ such that $P[j] = (t = t'')$ and $P[j'] = (t'' = t')$. By induction $P'[k_j] = \Big(\mathsf{d}(t,z) = \mathsf{d}(t'',z)\Big)$ and $P'[k_{j'}] = \Big(\mathsf{d}(t'',z') = \mathsf{d}(t',z')\Big)$. By using the substitution rule we can find a z'' that do not appear in any of t, t', and t'' and calculate $\Big(\mathsf{d}(t,z'') = \mathsf{d}(t'',z'')\Big)$ and $\Big(\mathsf{d}(t'',z'') = \mathsf{d}(t',z'')\Big)$. We accommodate this new calculations after $P'[k_{i-1}]$, define k_i as the next position, and calculate $P'[k_i] = \Big(\mathsf{d}(t,z'') = \mathsf{d}(t',z'')\Big)$ using transitivity.

If $P[i] = (t = t')$ was proven using symmetry, the proofs is more direct.

Suppose $P[i] = \Big(f(\xi_1,..,\xi_{\mathsf{r}(f)}), f(\xi'_1,..,\xi'_{\mathsf{r}(f)})\Big)$, with $f : \sigma_1 \ldots \sigma_{\mathsf{r}(f)} \to s$, was calculated using replacement. Then there are $j_1,\ldots,j_{\mathsf{r}f} < i$ such that $P[j_l] = (\xi_l = \xi'_l)$, for all $l \in \{j_1,\ldots,j_{\mathsf{r}f}\}$. By induction, $P'[k_{j_l}] = \Big(\mathsf{d}(\xi_l,z_l) = \mathsf{d}(\xi'_l,z_l)\Big)$, if $\sigma_l = s$ and $P'[k_{j_m}] = \Big(\mathfrak{d}(\xi_m,z_m) = \mathfrak{d}(\xi'_m,z_m)\Big)$, if $\sigma_m = d$ with $l,m \in \{j_1,\ldots,j_{\mathsf{r}f}\}$. By using the substitution rule we calculate $\Big(\mathsf{d}(\xi_l,xi'_l) = \mathsf{d}(\xi'_l,\xi'_l)\Big)$ and $\Big(\mathfrak{d}(\xi_m,\xi'_m) = \mathfrak{d}(\xi'_m,\xi'_m)\Big)$, and by axioms D1 and D3, and the transitivity rule, we have $\Big(\mathsf{d}(\xi_l,xi'_l) = 0\Big)$ and $\Big(\mathfrak{d}(\xi_m,\xi'_m) = 0\Big)$. We accommodate this new calculations after $P'[k_{i-1}]$, define k_i as the next position, and calculate $P'[k_i] = \Big(\mathsf{d}(f(\xi_1,..,\xi_{\mathsf{r}(f)}),z) = \mathsf{d}(f(\xi'_1,..,\xi'_{\mathsf{r}(f)}),z)\Big)$ using rule S1.

Suppose $P[i] = (\rho(t) = \rho(t'))$ was calculated using substitutivity. Then, there exists a $j < i$ such that $P[j] = (t = t')$. By induction $P'[k_j] = \Big(\mathsf{d}(t,z) = \mathsf{d}(t',z)\Big)$ with fresh z. Define the substitution ρ' by $\rho'(z) = y$ where y do not appear in $\rho(t)$ and $\rho(t')$ and $\rho'(x) = \rho(x)$ for any $x \neq z$. Use the substitutivity rule to calculate $P'[k_i] = \Big(\rho'(\mathsf{d}(t,z)) = \rho'(\mathsf{d}(t',z))\Big) = \Big(\mathsf{d}(\rho'(t),y)) = \mathsf{d}(\rho'(t'),y))\Big)$.

The cases of terms in sort d follows in the same manner. □

Corollary 6.27. *The following two statements hold:*

(a) For any state term $t \in \mathsf{T}(\Sigma)$ there exists $t' \in \mathsf{T}(\Sigma)$ in normal form such that $E^m_{CCS} \vdash d(t,z) = d(t',z)$.

(b) For any distribution term $\theta \in D\mathsf{T}(\Sigma)$ there exists $\theta' \in D\mathsf{T}(\Sigma)$ in normal form such that $E^m_{CCS} \vdash \mathfrak{d}(\theta,z) = \mathfrak{d}(\theta',z)$.

Proof. By Lemma 6.11 the equational theory E_{CCS} is normalizing and by using that terms are equal with its normal form the thesis follows then from Lemma 6.26. □

Now we can show that the equational theory E^m_{CCS} is also ground-complete for the bisimilarity metric $\mathbf{d}$.

Theorem 6.28. *E^m_{CCS} is ground-complete for the bisimilarity metric $\mathbf{d}$.*

Proof. We have to show that for any arithmetic expressions *expr* and *expr'* possibly containing closed terms of the form $\mathsf{d}(t,t')$ or $\mathfrak{d}(\theta,\theta')$ with $t,t' \in \mathsf{T}(\Sigma)$ and $\theta,\theta' \in \mathsf{DT}(\Sigma)$, we have $[\![expr]\!] = [\![expr']\!]$ implies $E^m_{CCS} \vdash expr = expr'$, i.e. $expr = expr'$ can be proved using axioms in E^m_{CCS} and arithmetic.

Therefore, we only need to show that $\mathbf{d}(t,t') = p$ implies $E^m_{CCS} \vdash \mathsf{d}(t,t') = p$ for all closed state terms $t,t' \in \mathsf{T}(\Sigma)$ and $p \in [0,1]$, and that $\mathbf{K}(\mathbf{d})([\![\theta]\!],[\![\theta']\!]) = p$ implies $E^m_{CCS} \vdash \mathfrak{d}(\theta,\theta') = p$ for all closed distribution terms $\theta,\theta' \in \mathsf{DT}(\Sigma)$. Since $\mathbf{d}(t,t')$ and $\mathbf{K}(\mathbf{d})([\![\theta]\!],[\![\theta']\!])$ are actual numbers in $[0,1]$, then we simply prove that $E^m_{CCS} \vdash \mathsf{d}(t,t') = \mathbf{d}(t,t')$ and $E^m_{CCS} \vdash \mathfrak{d}(\theta,\theta') = \mathbf{K}(\mathbf{d})([\![\theta]\!],[\![\theta']\!])$

By Corollary 6.27 and soundness of E^m_{CCS} for the bisimilarity metric $\mathbf{d}$ (Theorem 6.25), we can assume that t, t', θ, and θ' are in normal form. We proceed by induction in the height of the left term in the expressions $\mathsf{d}(t,t')$ and $\mathfrak{d}(\theta,\theta')$ supposing, by axioms D2 and D4, that it is less than or equal to the height of the corresponding right term.

We first calculate on state terms. For the base case $t = 0$. If $t' = 0$ then, by D1, $\mathrm{d}(0,0) = 0 = \mathbf{d}(0,0)$. If $t' = \sum_{i\in I} a_i.\theta_i$, then, by H1, $\mathrm{d}(0, \sum_{i\in I} a_i.\theta_i) = 1 = \mathbf{d}(0, \sum_{i\in I} a_i.\theta_i)$. For the inductive case, we calculate as follows:

$$\begin{aligned}
&\mathrm{d}\Big(\textstyle\sum_{i\in I} a_i.\theta_i, \sum_{j\in J} b_j.\theta'_j\Big)\\
=\ &\max\Big\{\max_{i\in I}\min_{j\in J, a_i=b_j} \lambda\cdot\mathfrak{d}(\theta_i,\theta'_j),\ \max_{j\in J}\min_{i\in I, a_i=b_j} \lambda\cdot\mathfrak{d}(\theta_i,\theta'_j)\Big\} && \text{(by axiom H2)}\\
=\ &\max\Big\{\max_{i\in I}\min_{j\in J, a_i=b_j} \lambda\cdot\mathbf{K}(\mathbf{d})(\theta_i,\theta'_j),\ \max_{j\in J}\min_{i\in I, a_i=b_j} \lambda\cdot\mathbf{K}(\mathbf{d})(\theta_i,\theta'_j)\Big\}\\
& && \text{(by induction hypothesis)}\\
=\ &\mathbf{d}\Big(\textstyle\sum_{i\in I} a_i.\theta_i, \sum_{j\in J} b_j.\theta'_j\Big)
\end{aligned}$$

Finally, the case of distribution terms is as follows

$$\begin{aligned}
&\mathfrak{d}\Big(\textstyle\bigoplus_{i\in I} p_i\delta(t_i), \bigoplus_{j\in J} q_j\delta(t'_j)\Big)\\
=\ &\min_{\omega\in\Omega(I,J)} \textstyle\sum_{i\in I, j\in J} \mathrm{d}(t_i,t'_j)\cdot\omega(i,j) && \text{(by axiom K)}\\
=\ &\min_{\omega\in\Omega(I,J)} \textstyle\sum_{i\in I, j\in J} \mathbf{d}(t_i,t'_j)\cdot\omega(i,j) && \text{(by induction hypothesis)}\\
=\ &\min_{\omega\in\Omega([\![\bigoplus_{i\in I} p_i\delta(t_i)]\!],[\![\bigoplus_{j\in J} q_j\delta(t'_j)]\!])} \textstyle\sum_{t,t'\in\mathsf{T}(\Sigma)} \mathbf{d}(t,t')\cdot\omega(t,t')\\
=\ &\mathbf{K}(\mathbf{d})([\![\textstyle\bigoplus_{i\in I} p_i\delta(t_i)]\!],[\![\bigoplus_{j\in J} q_j\delta(t'_j)]\!]) && \text{(by Definition 2.17)}
\end{aligned}$$

□

6.4.2 Axiomatization of bisimilarity metric of PGSOS

The algorithm of Figure 6.1 can be modified to provide axioms for bisimilarity metric for any operator defined in a generalized PGSOS specification as follows. Instead of adding the axioms in E_{CCS}, add the axioms in E^m_{CCS}, and for each equation $t_1 = t_2$ (resp. $\theta = \theta'$) added by the algorithm in Figure 6.1, add instead $\mathrm{d}(t_1,x) = \mathrm{d}(t_2,x)$ (resp. $\mathfrak{d}(\theta,\mu) = \mathfrak{d}(\theta',\mu)$).

Soundness of the axioms introduced by the algorithm is straightforward: we know that $t_1 \sim t_2$ implies $d(t_1,t_2) = 0$ and hence $d(t_1,t) = d(t_2,t)$ can be calculated from properties (ii) and (iii) in the definition of pseudometric (similarly for distribution terms).

We already observed that E^m_{CCS} is normalizing. Besides, it can be shown that any equational theory of the new algorithm is head-normalizing. Then, for every semantically well-founded closed term t there is a term t' in normal form such that $\mathrm{d}(t,t'') = \mathrm{d}(t',t'')$ for any t''. Using this elimination result ground-completeness follows.

Theorem 6.29 is the metric variant of the earlier Theorems 6.18–6.24 for bisimilarity equivalence.

Theorem 6.29. *Let P_i be a PTSS in generalized PGSOS format and let the PTSS P_o and the equational theory E_o be the outputs of the algorithm in Figure 6.1 modified as before. Then,*

(a) E_o is sound for the bisimilarity metric $\mathbf{d}$ in any disjoint extension of P_o, and

(b) E_o is ground-complete for the bisimilarity metric $\mathbf{d}$ in P_o, provided P_o is semantically well-founded.

Proof. We start by showing soundness for $\mathbf{d}$. The axioms added to E^m_{CCS} are all of the form $\mathrm{d}(t_1, x) = \mathrm{d}(t_2, x)$ (for each bisimilarity equivalence axiom $t_1 = t_2$), and $\mathfrak{d}(\theta_1, \mu) = \mathfrak{d}(\theta_2, \mu)$ (for each bisimilarity equivalence axiom $\theta_1 = \theta_2$). Soundness of these axioms is easy by the following argumentation. Let ρ be any closed substitution. First, observe that any equation $t_1 = t_2$ (resp. $\theta_1 = \theta_2$) implies bisimilarity $\rho(t_1) \sim \rho(t_2)$ (resp. $[\![\rho(\theta_1)]\!] \widetilde{\sim} [\![\rho(\theta_2)]\!]$) (Theorems 6.18–6.20 and the soundness part of Theorem 6.24). Then, because bisimilarity equivalence is the kernel of the bisimilarity metric $\mathbf{d}$ (Proposition 2.29), we get $\mathbf{d}(\rho(t_1), \rho(t_2)) = 0$ and $\mathbf{K}(\mathbf{d})([\![\rho(\theta_1)]\!], [\![\rho(\theta_2)]\!]) = 0$. Finally, by triangle inequality of the bisimulation metric $\mathbf{d}$ we get for the equation of state terms $t_1 = t_2$ that $\mathbf{d}(\rho(t_1), \rho(x)) \leq \mathbf{d}(\rho(t_1), \rho(t_2)) + \mathbf{d}(\rho(t_2), \rho(x)) = \mathbf{d}(\rho(t_2), \rho(x)) \leq \mathbf{d}(\rho(t_2), \rho(t_1)) + \mathbf{d}(\rho(t_1), \rho(x)) = \mathbf{d}(\rho(t_1), \rho(x))$. Hence, $\mathbf{d}(\rho(t_1), \rho(x)) = \mathbf{d}(\rho(t_2), \rho(x))$. The same reasoning applies also to the equations of distribution terms $\theta_1 = \theta_2$.

We proceed by showing ground-completeness. In detail, following the same argumentation as in the case of bisimulation equivalence (cf. proof of Theorem 6.24), we can show that the equational theory developed in Section 6.4 is head normalizing. This gives directly that for every semantically well-founded closed term t there is a term t' in normal form such that $\mathrm{d}(t, t'') = \mathrm{d}(t', t'')$ for any t''. Using this elimination result ground-completeness follows. □

6.5 Closing remarks

In this chapter we generalized the PGSOS format by defining a two-sorted signature that leads to a rigorous and clear definition of the distribution term in the target of positive literals. Moreover, this also fits nicely with the introduction of the equational theory. This carefully thought-out setting allows us to borrow the strategies of [ABV94] to obtain the algorithm of Figure 6.1 and prove its correctness (Theorem 6.24). This is particularly facilitated by the introduction of the operators mapping into sort d, and by the fact that all probabilistically lifted operators distribute with respect to $\oplus_p$ and δ.

We remark that the axiomatization E^m_{CCS} of bisimilarity metric is new in this paper. Axiom scheme H2 can be translated into a set of axioms that only include binary sum by introducing an auxiliary operator (Table 6.5). However we could not find so far a set of axioms that only use binary $\oplus_p$ operators in order to replace the axiom scheme K.

The presented algorithm can be generalized to behavioral equivalences weaker than bisimilarity equivalence (and resp. to behavioral metrics weaker than bisimilarity metric) if their respective equational theories contain E_{CCS} (resp. E^m_{CCS}), by following the approach of [GF13]. For instance, axiomatizing convex bisimilarity equivalence (introduced in [Seg95] under the name of probabilistic bisimilarity) is axiomatized by the equational theory $E_{CCS} \cup \{a.\mu_1 + a.\mu_2 = a.\mu_1 + a.\mu_2 + a.(\mu_1 \oplus_p \mu_2) \mid p \in \mathbb{Q} \cap (0,1)\}$. Moreover, the generalized PGSOS rule format needs a mild restriction developed in [DLG15b].

We used for the representation of probabilistic choices the (countably) infinite set of operators $\{\oplus_p | p \in \mathbb{Q} \cap (0,1)\}$ that are axiomatized by the equations P1–P3 given in Table 6.1. An alternative approach would be to use only a single operator $\oplus$ representing the probabilistic choice $\oplus_{0.5}$. Mean-value algebras developed by Heckmann [Hec94]

$$
\begin{aligned}
&\mathsf{d}(0,0) = 0 \\
&\mathsf{d}(0, a.\mu + x) = 1 \\
&\mathsf{d}(x,y) = \max\{\mathsf{ld}(x,y), \mathsf{ld}(y,x)\} \\
&\mathsf{ld}(x+y,z) = \max\{\mathsf{ld}(x,z), \mathsf{ld}(y,z)\} \\
&\mathsf{ld}(a.\mu, x+y) = \min\{\mathsf{ld}(a.\mu,x), \mathsf{ld}(a.\mu,y)\} \\
&\mathsf{ld}(a.\mu, b.\nu) = 1 \qquad \text{if } a \neq b \\
&\mathsf{ld}(a.\mu, a.\nu) = \lambda \cdot \mathfrak{d}(\mu,\nu)
\end{aligned}
\qquad
\begin{aligned}
&\mathsf{d}(x,y) = \max(\mathsf{mxd}(x,y) \cup \mathsf{mxd}(y,x)) \\
&\mathsf{mxd}(x+y,z) = \mathsf{mxd}(x,z) \cup \mathsf{mxd}(y,z) \\
&\mathsf{mxd}(0,x) = \emptyset \\
&\mathsf{mxd}(a.\mu,x) = \min(\mathsf{mnd}(a.\mu,x)) \\
&\mathsf{mnd}(a.\mu, x+y) = \mathsf{mnd}(a.\mu,x) \cup \mathsf{mnd}(a.\mu,y) \\
&\mathsf{mnd}(a.\mu, 0) = \{1\} \\
&\mathsf{mnd}(a.\mu, b.\nu) = \{1\} \qquad \text{if } a \neq b \\
&\mathsf{mnd}(a.\mu, a.\nu) = \{\lambda \cdot \mathfrak{d}(\mu,\nu)\}
\end{aligned}
$$

Table 6.5: Two axiomatizations of the Hausdorff-lifting using only binary summation.

provide an elegant equational axiomatization of this single probabilistic choice operator. This axiomatization has been recently lifted to metric spaces in [Bre+05]. Since some probabilistic choices such as $\oplus_{1/3}$ cannot be expressed as finite combination of $\oplus_{0.5}$ choices, the equational theory may need to be extended by adequate inequalities that approximate (arbitrarily close) those probabilistic choices. Along similar lines we suggest to investigate also if metric mean-value algebras and metric semilattices [Bre+05] allow for a primitive formulation of the equations H2 and K in Table 6.4 (Hausdorff and Kantorovich lifting operators). We leave the further development of the technical details as future work.

Another interesting research direction is to investigate if the axiomatizations provided in this chapter give rise to a term rewriting system with "good" properties such as normalisation and confluence. This question has been investigated for the nondeterministic setting in [Bos94] and [GF13, Section 4]. The axiomatization of a behavioral equivalence or behavioral metric gives rise to a term rewriting system with rewrite rules that are directed versions of the equations. Moreover, we need to exclude associativity and commutativity to obtain termination, and add some auxiliary rules to obtain confluence [Bos94]. Preliminary results suggest that this approach gives also in the probabilistic setting a head normalising and confluent term rewriting systems (but not strongly normalising since already Bosscher's original rulified axiomatisation is not strongly normalizing). Finally, the properties of the term rewriting system arising from the axiomatization of behavioral equivalences and behavioral metrics of specifications that combine both SOS rules and non-structural assignment rules [MR05; RB14] are of interest. The interesting question is which properties the non-structural assignment rules guarantee that the resulting term rewriting system is still normalizing and/or confluent.

Chapter 7

Conclusions

Modern information and computing systems are distributed and concurrent. There are three important aspects that determine the specification and verification of those systems: a model of the system, a language to describe the system, and a behavioral semantics that assigns a meaning to the system. For distributed and concurrent systems not only functional correctness but also quantitative properties such as reliability, performance and resources play a central role for both modeling of the system and also analysis and verification of the system properties. Probability is a well-understood quantitative property that allows us to model and measure reliability, availability, serviceability, security and trust.

Labeled transition systems are the common model to describe the operational semantics of a system in terms of single-step executions. Transition labels denote the different actions a system can execute and provide a natural means to formalize various notions of synchronous and asynchronous composition. Structural operational semantics is a widely used and accepted formal approach to specify the operational semantics of programming languages and process algebras in terms of deduction rules that describe the single-step execution of the system. The deduction rules describe in a structural compositional manner the operational behavior of the system. Rule and specification formats specify classes of rules and specifications where the structural composition of states and the respective operational behavior preserves important properties of the behavioral semantics.

Labelled transition systems have been extended to Probabilistic labelled transition systems to model probabilistic choices that may occur in the single-step execution of the underlying system [Seg95]. Similarly, also the structural operational semantics approach has been extended to allow for the specification of languages that include probabilistic choices [DGL15]. The careful design of the probabilistic specification theory, especially the separation between probabilistic and nondeterministic choice as initiated in [Bar04] and [DL12], allowed us to develop smoothly further the specification meta-theory of probabilistic languages.

The research in this thesis is motivated by the observation that the well-understood bisimulation equivalence semantics is not sufficient for probabilistic nondeterministic transition systems. The recently developed bisimulation metric semantics is a robust

semantic notion to define the behavioral distance between systems. The bisimulation distance between two states of a system (or two systems when considering the distance between their initial states) quantifies the proximity of their respective behavioral properties. However, considering bisimulation metric semantics there have been neither sufficient notions of compositionality of probabilistic systems nor satisfying operational, denotational and axiomatic specification approaches have been developed so far.

Centered around the main research question how to specify probabilistic languages and reason over probabilistic systems w.r.t. bisimulation metric semantics we defined appropriate concepts and notions of compositionality, formalized specification approaches of compositional languages, formalized in terms of a denotational model the primitive process behavior determining the compositionality of operators, and axiomatized the algebraic properties of operators. These results together provide a well-rounded contribution to the theory of formal languages, automata theory and semantics.

We started in chapter 3 by examining the first research question Q1 (page 7) and developed the concept of uniformly continuous operators that captures the essence of compositional reasoning. The modulus of continuity formalizes the intuitive notion that replacing components by similar components the composed system is still (quantifiably) similar (with similarity quantified by the bisimulation distance). We analyzed the compositionality properties of many process algebra operators and observed that the compositionality property of non-expansiveness advocated in earlier research is not sufficient for recursive operators.

We proceeded then in chapter 4 by examining the second research question Q2 (page 8) and proposed appropriate SOS rule and specification formats that allow us to formally define operators for any given modulus of continuity. Our main insight is that the replication of processes and the probabilistic choices of operators determines their compositionality property. Moreover, these structural properties compose in a natural way and allow us to determine the modulus of continuity (understood as its semantical compositionality property) in a compositional way from the syntactical specification of that operator.

As a next step we developed in chapter 5 a denotational model that characterizes the primitive process behavior (replication, probabilistic choice, nondeterministic choice) that determines its compositionality property. This chapter answers the third research question Q3 (page 9) and refines numerous results of the former chapters by defining a fixed-point calculus that computes compositionally denotation of operators and open terms by mimicking both their syntactic structure and the operational behavior. The denotational model makes also explicit how the primitive process behavior of operators (replication, probabilistic choice, nondeterministic choice) interact. Hence, the denotational approach gives system and language designers a tool to define languages with the best trade-off (w.r.t. the application context) between the different primitive process behavior.

In the last chapter 6 we studied the axiomatization of language expressions to answer the last research question Q4 (page 10). We provided an algorithm that produces for both bisimilarity equivalence and bisimilarity metric and equational theory that is sound and ground-complete. We generalized the PGSOS format by formalizing states and distributions as separate sorts and that also allows operators to take distributions as arguments. This makes the action prefix operator (well-known from process algebra) a single operator with a single rule (and does not require the countably infinite set of operators and

rules with the former PGSOS format).

The technical chapters 3–6 answer thoroughly the research questions Q1–Q4 raised in the introduction. The theoretical concepts, practical tools and conceptual methods developed on those chapters form collectively a well-elaborated theory to formally specify and reason over probabilistic processes and programs. Practically, language designers may use the theory to specify language operators that will be used by engineers to code and applied by operators to describe programs and systems that behave smoothly also in cases of external or internal disturbance. Scientifically, the concepts, approaches and results in this thesis open the door to generalize the specification theories, modeling approaches, proof-theory and model-checking of quantitative formalisms (e.g. continuous time systems, weighted automata, quantitative games) to a robust semantic notion based on behavioral metrics and metric compositionality.

Future work Each chapter provided multiple directions for future work which we will just summarize here. In this thesis we focused on behavioral metric semantics. A natural continuation is to explore the research questions and analyze the developed approaches in the context of other behavioral metrics, such as trace, testing and weak semantics. Trace and testing semantics are important notions of behavioral semantics that are less strict than bisimulation semantics and measure the distance between states in term of execution traces and probing tests of external observers. Those semantic notions become relevant if the system designer is for instance only interested in linear properties of the system. Moreover, notions of weak semantics are essential to effectively reason over the refinement of systems, esp. for action refinement and for abstracting from and hiding of internal behavior. Additionally, in order to specify and analyze quantitative properties such as time and resources the research questions and results should be reconsidered by changing the underlying model of probabilistic nondeterministic transition systems to Markov automata and weighted automata.

We focussed in this thesis mainly on the specification of languages and systems. This research should be follow-up by investigating also system verification and model-checking when considering behavioral metric semantics. Important research questions that arise are which quantitative logics characterize trace, test and weak notions of behavioral metric semantics. Moreover, the interpretation of the behavioral distances, their relation to natural system properties and the application in performance validation (understood as the metric extension of binary correctness validation) are pressing questions that should be investigated. Finally, we propose to investigate how our results focussing on the specification of operational semantics in terms of deductive reasoning may be applied to languages of probabilistic programming that provide a probabilistic abductive reasoning and follow the Bayesian inference principle.

Bibliography

[ABV94] Luca Aceto, Bard Bloom and Frits Vaandrager. 'Turning SOS rules into equations'. In: *I&C* 111.1 (1994), pp. 1–52.

[AFV01a] Luca Aceto, Wan Fokkink and Chris Verhoef. 'Conservative Extension in Structural Operational Semantics'. In: *Current Trends in Theoretical Computer Science*. 2001, pp. 504–524.

[AFV01b] Luca Aceto, Wan Fokkink and Chris Verhoef. 'Structural operational semantics'. In: *Handbook of Process Algebra*. Elsevier, 2001, pp. 197–292.

[Ace+13] Luca Aceto et al. 'Exploiting Algebraic Laws to Improve Mechanized Axiomatizations'. In: *Proc. CALCO'13*. Vol. 8089. LNCS. Springer, 2013, pp. 36–50.

[Alf97] Luca de Alfaro. 'Formal verification of probabilistic systems'. PhD thesis. Stanford University, 1997.

[AFS04] Luca de Alfaro, Marco Faella and Mariëlle Stoelinga. 'Linear and Branching Metrics for Quantitative Transition Systems'. In: *Proc. ICALP'04*. Vol. 3142. LNCS. Springer, 2004, pp. 97–109.

[AHM03] Luca de Alfaro, Thomas A. Henzinger and Rupak Majumdar. 'Discounting the Future in Systems Theory'. In: *Proc. ICALP'03*. Springer, 2003, pp. 1022–1037.

[Alf+07] Luca de Alfaro et al. 'Game relations and metrics'. In: *Proc. LICS'07*. IEEE. 2007, pp. 99–108.

[Alf+08] Luca de Alfaro et al. 'Game Refinement Relations and Metrics'. In: *LMCS* 4.3 (2008).

[And99] Suzana Andova. 'Process algebra with probabilistic choice'. In: *Proc. ARTS'99*. Vol. 1601. LNCS. Springer, 1999, pp. 111–129.

[And02] Suzana Andova. 'Probabilistic process algebra'. PhD thesis. Eindhoven University of Technology, 2002.

[Arn94] André Arnold. *Finite Transition Systems - Semantics of Communicating Systems*. Prentice Hall, 1994.

[Arn+14] Florian Arnold et al. 'A Tutorial on Interactive Markov Chains'. In: *Stochastic Model Checking. Rigorous Dependability Analysis Using Model Checking Techniques for Stochastic Systems*. Vol. 8453. LNCS. Springer, 2014, pp. 26–66.

[Bac+13] Giorgio Bacci et al. ‘Computing Behavioral Distances, Compositionally’. In: *Proc. MFCS’13*. Vol. 8087. LNCS. Springer, 2013, pp. 74–85.

[Bac+15] Giorgio Bacci et al. ‘Converging from Branching to Linear Metrics on Markov Chains’. In: *Proc. ICTAC’15*. Vol. 9399. LNCS. Springer, 2015, pp. 1–19.

[BBR10] Jos C. M. Baeten, Twan Basten and M. A. Reniers. *Process algebra: equational theories of communicating processes*. Vol. 50. Cambridge University Press, 2010.

[BBK87] Jos C. M. Baeten, Jan A. Bergstra and Jan Willem Klop. ‘On the consistency of Koomen’s Fair Abstraction Rule’. In: *TCS* 51 (1987), pp. 129–176.

[Bai+04] Christel Baier et al. *Validation of stochastic systems: A guide to Current Research*. Vol. 2925. LNCS. Springer, 2004.

[BS01] E. Bandini and R. Segala. ‘Axiomatizations for Probabilistic Bisimulation’. In: *Proc. ICALP’01*. Vol. 2076. LNCS. Springer, 2001, pp. 370–381.

[Bar02] Falk Bartels. ‘GSOS for probabilistic transition systems’. In: *Proc. CMCS’02*. Vol. 65. ENTCS. Elsevier, 2002, pp. 29–53.

[Bar04] Falk Bartels. ‘On Generalised Coinduction and Probabilistic Specification Formats’. PhD thesis. VU University Amsterdam, 2004.

[BDL13] Marco Bernardo, Rocco De Nicola and Michele Loreti. ‘The Spectrum of Strong Behavioral Equivalences for Nondeterministic and Probabilistic Processes’. In: *Proc. QAPL’13*. Vol. 117. EPTCS. 2013, pp. 81–96.

[BFG04] Bard Bloom, Wan Fokkink and Rob J. van Glabbeek. ‘Precongruence formats for decorated trace semantics’. In: *ACM TOCL* 5 (1 2004), pp. 26–78.

[BIM95] Bard Bloom, Sorin Istrail and Albert R. Meyer. ‘Bisimulation can’t be traced’. In: *J. ACM* 42.1 (1995), pp. 232–268.

[BG96] Roland Bol and Jan Friso Groote. ‘The meaning of negative premises in transition system specifications’. In: *J. ACM* 43 (5 1996), pp. 863–914.

[Bos94] Doeko J. B. Bosscher. ‘Term rewriting properties of SOS axiomatisations’. In: *Proc. TACS’94*. Springer. 1994, pp. 425–439.

[Bre12] Franck van Breugel. ‘On behavioural pseudometrics and closure ordinals’. In: *Information Processing Letters* 112.19 (2012), pp. 715–718.

[BW01] Franck van Breugel and James Worrell. ‘Towards Quantitative Verification of Probabilistic Transition Systems’. In: *Proc. ICALP’01*. Vol. 2076. LNCS. Springer, 2001, pp. 421–432.

[BW05] Franck van Breugel and James Worrell. ‘A Behavioural Pseudometric for Probabilistic Transition Systems’. In: *TCS* 331.1 (2005), pp. 115–142.

[Bre+05] Franck van Breugel et al. ‘An Accessible Approach to Behavioural Pseudometrics’. In: *Proc. ICALP’05*. Vol. 3580. LNCS. Springer, 2005, pp. 1018–1030.

[BHR84] Stephen D. Brookes, Charles A. R. Hoare and Andrew W. Roscoe. ‘A theory of communicating sequential processes’. In: *JACM* 31.3 (1984), pp. 560–599.

[Cha+14] Konstantinos Chatzikokolakis et al. 'Generalized bisimulation metrics'. In: *Proc. CONCUR'14*. Vol. 8704. LNCS. Springer, 2014, pp. 32–46.

[CMR97] Evelyne Contejean, Claude Marché and Landy Rabehasaina. 'Rewrite Systems for Natural, Integral, and Rational Arithmetic'. In: *Proc. RTA'97*. Vol. 1232. LNCS. Springer, 1997, pp. 98–112.

[CR11] Silvia Crafa and Francesco Ranzato. 'A spectrum of behavioral relations over LTSs on probability distributions'. In: *Proc. CONCUR'11*. Vol. 6901. LNCS. Springer, 2011, pp. 124–139.

[DGL14] Pedro R. D'Argenio, Daniel Gebler and Matias D. Lee. 'Axiomatizing Bisimulation Equivalences and Metrics from Probabilistic SOS Rules'. In: *Proc. FoSSaCS'14*. Vol. 8412. LNCS. Springer, 2014, pp. 289–303.

[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.

[DL12] Pedro R. D'Argenio and Matias D. Lee. 'Probabilistic Transition System Specification: Congruence and Full Abstraction of Bisimulation'. In: *Proc. FoSSaCS'12*. Vol. 7213. LNCS. Springer, 2012, pp. 452–466.

[DLG15a] Pedro R. D'Argenio, Matias D. Lee and Daniel Gebler. 'SOS rule formats for convex and abstract probabilistic bisimulations'. In: *Proc. EXPRESS/SOS'15*. Vol. 190. EPTCS. 2015, pp. 31–45.

[DLG15b] Pedro R. D'Argenio, Matias David Lee and Daniel Gebler. 'SOS rule formats for convex and abstract probabilistic bisimulations'. In: *Proc. EXPRESS/SOS'15*. Vol. 190. EPTCS. 2015, pp. 31–45.

[DP02] Brian A. Davey and Hilary A. Priestley. *Introduction to lattices and order*. Cambridge University Press, 2002.

[Deh+14] Christian Dehnert et al. 'On Abstraction of Probabilistic Systems'. In: *Stochastic Model Checking. Rigorous Dependability Analysis Using Model Checking Techniques for Stochastic Systems*. Vol. 8453. LNCS. Springer, 2014, pp. 87–116.

[Den15] Yuxin Deng. *Semantics of Probabilistic Processes: An Operational Approach*. Springer, 2015.

[DD07] Yuxin Deng and Wenjie Du. 'Probabilistic Barbed Congruence'. In: *Proc. QAPL'07*. Vol. 190. ENTCS 3. 2007, pp. 185–203.

[DD09] Yuxin Deng and Wenjie Du. 'The Kantorovich Metric in Computer Science: A Brief Survey'. In: *Proc. QAPL'09*. Vol. 253. ENTCS 3. 2009, pp. 73–82.

[DD11] Yuxin Deng and Wenjie Du. *Logical, Metric, and Algorithmic Characterisations of Probabilistic Bisimulation*. Tech. rep. CMU-CS-11-110. CMU, Mar. 2011.

[Den+05] Yuxin Deng et al. 'Metrics for Action-labelled Quantitative Transition Systems'. In: *Proc. QAPL'05*. Vol. 153. EPTCS 2. 2005, pp. 79–96.

[Den+07] Yuxin Deng et al. 'Remarks on testing probabilistic processes'. In: *ENTCS* 172 (2007). Computation, Meaning, and Logic: Articles dedicated to Gordon Plotkin, pp. 359–397.

[DLT08] Josée Desharnais, Francois Laviolette and Mathieu Tracol. 'Approximate Analysis of Probabilistic Processes: Logic, Simulation and Games'. In: *Proc. QEST'08*. IEEE. 2008, pp. 264–273.

[Des+02a] Josée Desharnais et al. 'The Metric Analogue of Weak Bisimulation for Probabilistic Processes'. In: *Proc. LICS'02*. IEEE. 2002, pp. 413–422.

[Des+02b] Josée Desharnais et al. 'Weak Bisimulation is Sound and Complete for PCTL*'. In: *Proc. CONCUR'02*. Vol. 2421. LNCS. Springer, 2002, pp. 355–370.

[Des+04] Josée Desharnais et al. 'Metrics for Labelled Markov Processes'. In: *TCS* 318.3 (2004), pp. 323–354.

[FL14] Uli Fahrenberg and Axel Legay. 'The quantitative linear-time-branching-time spectrum'. In: *TCS* 538 (2014), pp. 54–69.

[Fel08] Willliam Feller. *An Introduction to Probability Theory and Its Applications*. Vol. 2. John Wiley & Sons, 2008.

[Fok07] Wan Fokkink. *Modelling Distributed Systems*. Springer, 2007.

[Fok13] Wan Fokkink. *Introduction to Process Algebra*. Springer, 2013.

[FGW06a] Wan Fokkink, Rob J. van Glabbeek and Paulien de Wind. 'Compositionality of Hennessy-Milner logic by structural operational semantics'. In: *TCS* 354 (3 2006), pp. 421–440.

[FGW06b] Wan Fokkink, Rob J. van Glabbeek and Paulien de Wind. 'Divide and Congruence Applied to η-Bisimulation'. In: *Proc. SOS'05*. Vol. 156. ENTCS. Elsevier, 2006, pp. 97–113.

[FGW06c] Wan Fokkink, Rob J. van Glabbeek and Paulien de Wind. 'Divide and Congruence: From Decomposition of Modalities to Preservation of Branching Bisimulation'. In: *Proc. FMCO'05*. Vol. 4111. LNCS. Springer, 2006, pp. 195–218.

[FGW12] Wan Fokkink, Rob J. van Glabbeek and Paulien de Wind. 'Divide and congruence: From decomposition of modal formulas to preservation of branching and η-bisimilarity'. In: *I&C* 214 (2012), pp. 59–85.

[GF10] Maciej Gazda and Wan Fokkink. 'Congruence from the Operator's Point of View: Compositionality Requirements on Process Semantics'. In: *Proc. SOS'10*. Vol. 32. EPTCS. 2010, pp. 15–25.

[GF13] Maciej Gazda and Wan Fokkink. 'Turning GSOS Rules into Equations for Linear Time-Branching Time Semantics'. In: *The Computer Journal* 56.1 (2013), pp. 34–44.

[GF12] Daniel Gebler and Wan Fokkink. 'Compositionality of Probabilistic Hennessy-Milner Logic through Structural Operational Semantics'. In: *Proc. CONCUR'12*. Vol. 7454. LNCS. Springer, 2012, pp. 395–409.

[GGM13] Daniel Gebler, Eugen-Ioan Goriac and Mohammad Reza Mousavi. 'Algebraic Meta-Theory of Processes with Data'. In: *Proc. EXPRESS/SOS'13*. Vol. 120. EPTCS. 2013, pp. 63–77.

[GHT14] Daniel Gebler, Vahid Hashemi and Andrea Turrini. 'Computing Behavioral Relations for Probabilistic Concurrent Systems'. In: *Stochastic Model Checking. Rigorous Dependability Analysis Using Model Checking Techniques for Stochastic Systems*. Vol. 8453. LNCS. Springer, 2014, pp. 117–155.

[GLT15] Daniel Gebler, Kim G. Larsen and Simone Tini. 'Compositional metric reasoning with Probabilistic Process Calculi'. In: *Proc. FoSSaCS'15*. Vol. 9034. LNCS. Springer, 2015, pp. 230–245.

[GT13] Daniel Gebler and Simone Tini. 'Compositionality of Approximate Bisimulation for Probabilistic Systems'. In: *Proc. EXPRESS/SOS'13*. Vol. 120. EPTCS. 2013, pp. 32–46.

[GT14] Daniel Gebler and Simone Tini. 'Fixed-point Characterization of Compositionality Properties of Probabilistic Processes Combinators'. In: *Proc. EXPRESS/SOS'14*. Vol. 160. EPTCS. 2014, pp. 63–78.

[GT15] Daniel Gebler and Simone Tini. 'SOS Specifications of Probabilistic Systems by uniformly continuous operators'. In: *Proc. CONCUR'15*. Vol. 42. LIPIcs, 2015, pp. 155–168.

[GJS90] Alessandro Giacalone, Chi-Chang Jou and Scott A. Smolka. 'Algebraic Reasoning for Probabilistic Concurrent Systems'. In: *Proc. IFIP TC2 Working Conf. on Prog. Concepts and Methods*. 1990, pp. 443–458.

[Gla90] Rob J. van Glabbeek. 'The Linear Time - Branching Time Spectrum I'. In: *CONCUR'90*. Vol. 458. LNCS. Springer, 1990, pp. 278–297.

[Gla93] Rob J. van Glabbeek. 'The Linear Time - Branching Time Spectrum II'. In: *CONCUR'93*. Vol. 715. LNCS. Springer, 1993, pp. 66–81.

[GSS95] Rob J. van Glabbeek, Scott A. Smolka and Bernhard Steffen. 'Reactive, Generative, and Stratified Models of Probabilistic Processes'. In: *I&C* 121.1 (1995), pp. 59–80.

[GM82] Joseph A. Goguen and José Meseguer. 'Completeness of many-sorted equational logic'. In: *SIGPLAN Notices* 17.1 (Jan. 1982), pp. 9–17.

[Gor+14] Andrew D. Gordon et al. 'Probabilistic programming'. In: *Proc. ICSE'14*. ACM. 2014, pp. 167–181.

[Gro93] Jan Friso Groote. 'Transition System Specifications with Negative Premises'. In: *TCS* 118.2 (1993), pp. 263–299.

[GV92] Jan Friso Groote and Frits Vaandrager. 'Structured Operational Semantics and Bisimulation as a Congruence'. In: *I&C* 100.2 (1992), pp. 202–260.

[Hal60] Paul R. Halmos. *Naive set theory*. Springer, 1960.

[Han94] Hans A. Hansson. *Time and Probability in Formal Design of Distributed Systems*. Ed. by Lars-Ake Fredlund. New York, NY, USA: Elsevier, 1994.

[HJ94] Hans Hansson and Bengt Jonsson. 'A logic for reasoning about time and reliability'. In: *FAC* 6.5 (1994), pp. 512–535.

[Hav01] Boudewijn R. Haverkort. 'Markovian models for performance and dependability evaluation'. In: *LNCS* 2090 (2001), pp. 38–83.

[Hec94] Reinhold Heckmann. 'Probabilistic domains'. In: *Proc. CAAP'94*. Vol. 787. LNCS. Springer, 1994, pp. 142–156.

[Hen88] Matthew Hennessy. *Algebraic theory of processes*. MIT press, 1988.

[Hen12] Matthew Hennessy. 'Exploring probabilistic bisimulations, part I'. In: *FAC* 24.4-6 (2012), pp. 749–768.

[Hen13] Thomas A. Henzinger. 'Quantitative reactive modeling and verification'. In: *Computer Science - R&D* 28.4 (2013), pp. 331–344.

[HS06] Thomas A. Henzinger and Joseph Sifakis. 'The embedded systems design challenge'. In: *Proc. FM'06*. Vol. 4085. LNCS. Springer, 2006, pp. 1–15.

[Her+11] Holger Hermanns et al. 'Probabilistic Logical Characterization'. In: *I&C* 209.2 (2011), pp. 154–172.

[Hoa85] C. Antony R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[HLW08] Bernhard Hoffman-Wellenhof, Herbert Lichtenegger and Elmar Wasle. *GNSS-Global Navigation Satellite Systems: GPS, GLONASS, Galileo and more*. Springer, 2008.

[JL91] Bengt Jonsson and Kim G. Larsen. 'Specification and refinement of probabilistic processes'. In: *Proc. LICS'91*. IEEE. 1991, pp. 266–277.

[JLY01] Bengt Jonsson, Kim G. Larsen and Wang Yi. 'Probabilistic Extensions of Process Algebras'. In: *Handbook of Process Algebra*. Elsevier, 2001, pp. 685–710.

[Kal06] Olav Kallenberg. *Foundations of modern probability*. Springer, 2006.

[KBL01] Joost-Pieter Katoen, Christel Baier and Diego Latella. 'Metric semantics for true concurrent real time'. In: *TCS* 254.1–2 (2001), pp. 501–542.

[Kel76] Robert M. Keller. 'Formal Verification of Parallel Programs'. In: *Commun. ACM* 19.7 (1976), pp. 371–384.

[KS60] John G. Kemeny and James Laurie Snell. *Finite Markov Chains*. Van Nostrand, Princeton, N.J., 1960.

[Kli05] Bartek Klin. 'From bialgebraic semantics to congruence formats'. In: *Proc. SOS'04*. Vol. 128. ENTCS 1. Elsevier, 2005, pp. 3–37.

[Kli09] Bartek Klin. 'Bialgebraic methods and modal logic in structural operational semantics'. In: *I&C* 207 (2 2009), pp. 237–257.

[Kli10] Bartek Klin. 'Structural Operational Semantics and Modal Logic, Revisited'. In: *Proc. CMCS'10*. Vol. 264. ENTCS 2. Elsevier, 2010, pp. 155–175.

[KS13] Bartek Klin and Vladimiro Sassone. 'Structural operational semantics for stochastic and weighted transition systems'. In: *I&C* 227 (2013), pp. 58–83.

[LT05] Ruggero Lanotte and Simone Tini. 'Probabilistic Congruence for Semistochastic Generative Processes'. In: *Proc. FoSSaCS'05*. Vol. 3441. LNCS. Springer, 2005, pp. 63–78.

[LT09] Ruggero Lanotte and Simone Tini. 'Probabilistic Bisimulation as a Congruence'. In: *ACM TOCL* 10 (2 2009), pp. 1–48.

[LS91] Kim G. Larsen and Arne Skou. 'Bisimulation Through Probabilistic Testing'. In: *I&C* 94 (1 1991), pp. 1–28.

[LX91] Kim G. Larsen and Liu Xinxin. 'Compositionality through an Operational Semantics of Contexts'. In: *J. Log. Comput.* 1.6 (1991), pp. 761–795.

[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*. Vol. 89. EPTCS. 2012, pp. 115–130.

[LV15] Matias D. Lee and Erik P. de Vink. 'Rooted branching bisimulation as a congruence for probabilistic transition systems'. In: *Proc. QAPL'15*. Vol. 194. EPTCS. 2015, pp. 79–94.

[Mar+14] Radu Mardare et al. 'Continuity Properties of Distances for Markov Processes'. In: *Proc. QEST'14*. IEEE. 2014, pp. 297–312.

[Mil80] Robin Milner. *A calculus of communicating systems*. Springer, 1980.

[Mil89] Robin Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[Mio14] Matteo Mio. 'Upper-Expectation Bisimilarity and Łukasiewicz μ-Calculus'. In: *Proc. FoSSaCS'14*. Vol. 8412. LNCS. Springer, 2014, pp. 335–350.

[MS13] Matteo Mio and Alex Simpson. 'A Proof System for Compositional Verification of Probabilistic Concurrent Processes'. In: *Proc. FoSSaCS'13*. Vol. 7794. LNCS. Springer, 2013, pp. 161–176.

[MR05] Mohammad Reza Mousavi and Michel A. Reniers. 'Congruence for Structural Congruences'. In: *Proc. FoSSaCS'05*. Vol. 3441. LNCS. Springer, 2005, pp. 47–62.

[MRG07] Mohammad Reza Mousavi, Michel A. Reniers and Jan Friso Groote. 'SOS formats and meta-theory: 20 years after'. In: *TCS* 373.3 (2007), pp. 238–272.

[Pan09] Prakash Panangaden. *Labelled Markov Processes*. Imperial College Press, 2009.

[Par81] David Park. 'Concurrency and Automata on Infinite Sequences'. In: *Proc. 5th GI-Conference on TCS*. Vol. 104. LNCS. Springer, 1981, pp. 167–183.

[PS07] Augusto Parma and Roberto Segala. 'Logical Characterizations of Bisimulations for Discrete Probabilistic Systems'. In: *Proc. FoSSaCS'07*. Vol. 4423. LNCS. Springer, 2007, pp. 287–301.

[Plo81] Gordon Plotkin. *A Structural Approach to Operational Semantics*. Report DAIMI FN-19. Reprinted in JLAP, 60-61:17-139, 2004. Aarhus University, 1981.

[Ram10] Vishwanath Raman. 'Game Relations, Metrics and Refinements'. PhD thesis. University of California at Santa Cruz, 2010.

[RS14] Anne Remke and Mariëlle Stoelinga. *Stochastic Model Checking. Rigorous Dependability Analysis Using Model Checking Techniques for Stochastic Systems*. Vol. 8453. Springer, 2014.

[RB14] Jurriaan Rot and Marcello Bonsangue. 'Combining Bialgebraic Semantics and Equations'. In: *Proc. FoSSaCS'14*. Vol. 8412. LNCS. Springer. 2014, pp. 381–395.

[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'. In: *Nordic J. of Computing* 2 (2 1995), pp. 250–273.

[Sim84] Robert de Simone. 'Calculabilité et expressivité dans l'algèbre de processus parallèles Meije'. PhD thesis. Univ. Paris 7, 1984.

[Sim85] Robert de Simone. 'Higher-Level Synchronising Devices in Meije-SCCS'. In: *TCS* 37 (1985), pp. 245–267.

[SDC07] Lin Song, Yuxin Deng and Xiaojuan Cai. 'Towards Automatic Measurement of Probabilistic Processes'. In: *Proc. QSIC'07*. IEEE. 2007, pp. 50–59.

[SC05] Daniel H. Steinberg and Stuart Cheshire. *Zero Configuration Networking: The Definitive Guide: The Definitive Guide*. O'Reilly Media, Inc., 2005.

[Ste94] William J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.

[Sto02] Mariëlle Stoelinga. 'An Introduction to Probabilistic Automata'. In: *Bulletin of the EATCS* 78 (2002), pp. 176–198.

[Tin08] Simone Tini. 'Non Expansive ϵ-bisimulations'. In: *Proc. AMAST'08*. Vol. 5140. LNCS. Springer, 2008, pp. 362–376.

[Tin10] Simone Tini. 'Non-expansive ϵ-bisimulations for Probabilistic Processes'. In: *TCS* 411 (22-24 2010), pp. 2202–2222.

[TDZ11] Mathieu Tracol, Josée Desharnais and Abir Zhioua. 'Computing Distances between Probabilistic Automata'. In: *Proc. QAPL'11*. Vol. 57. EPTCS. 2011, pp. 148–162.

[Vil08] Cédric Villani. *Optimal transport: old and new*. Vol. 338. Springer, 2008.

[Yin02a] Mingsheng Ying. 'Additive models of probabilistic processes'. In: *TCS* 275.1 (2002), pp. 481–519.

[Yin02b] Mingsheng Ying. 'Bisimulation indexes and their applications'. In: *TCS* 275.1 (2002), pp. 1–68.

[Yux15] Daniel Gebler Yuxin Deng Wenjie Du. 'Modal Characterisations of Behavioural Pseudometrics'. Under submission. 2015.

[ZZ08] Jinjin Zhang and Zhaohui Zhu. 'A Behavioural Pseudometric based on λ–Bisimilarity'. In: *ENTCS* 220.3 (2008), pp. 115–127.

Summary

Robust SOS Specifications of Probabilistic Processes

We develop a language specification theory for probabilistic nondeterministic systems in the context of metric bisimulation semantics. This theory generalizes and extends ordinary Plotkin-style Structural Operational Semantics language specification theory of nondeterministic systems. The main contributions are:

- unification and generalization of existing metric compositionality concepts
- language specification formats for all metric compositionality concepts
- language meta-theoretical results for metric compositional language specifications
- denotational model of metric compositionality of language operators and programs
- axiomatization of the bisimulation distance of language operators and programs

We introduce the notion of uniformly continuous language operators as canonical metric compositionality concept that generalizes earlier proposals like non-extensive and non-expansive operators and allows us now to reason also about recursive programs. Then we develop a spectrum of language specification formats for the spectrum of compositionality properties formed by uniform continuity, Lipschitz continuity, non-expansiveness and non-extensiveness that allows us to specify simultaneously unbounded recursive, bounded recursive, non-recursive and choice operators. We derive for each language specification format compositionality results that allow us to determine the distance between programs and provide methods to verify if a program specification satisfies some compositionality property. Complementary to the structural specification results we develop a denotational model that formalizes in a stratified manner the primitive program behavior that determines the compositionality property of language operators. Finally, an equational axiomatization of the behavioral distance of language operators derived from their respective specifications provides an algorithmic approach to deduce algebraic properties such as the distance between programs.

Samenvatting

Robuuste SOS Specificaties van Probabilistische Processen

In dit proefschrift wordt een theorie ontwikkeld voor specificatie-talen van nietdeterministische systemen in de context van metrische bisimulatie-semantiek. Deze theorie generaliseert en is een uitbreiding van de door Plotkin geïntroduceerde structurele operationele semantiek. De belangrijkste bijdragen zijn:

- generalisatie van bestaande concepten voor metrische compositionaliteit
- formaten voor taal-specificatie met betrekking tot al deze concepten
- meta-theoretische resultaten voor metrische compositionele taal-specificaties
- een denotationeel model van metrische compositionaliteit voor taal-operatoren en programma's
- axiomatisatie van bisimulatie-afstand voor taal-operatoren en programma's

We introduceren de notie van uniform continue taal-operatoren als een canoniek concept voor metrische compositionaliteit dat eerdere voorstellen zoals niet-extensieve en niet-expansieve operatoren generaliseert en het mogelijk maakt om te redeneren over recursieve programma's. Vervolgens ontwikkelen we een spectrum van formaten voor taalspecificatie voor het spectrum van compositionaliteitseigenschappen gevormd door uniforme continuïteit, niet-expansiviteit en niet-extensiviteit dat het mogelijk maakt om simultaan onbegrensd recursieve, begrensd recursieve, niet-recursieve en keuze-operatoren te specificeren. We bewijzen voor ieder van deze formaten compositionaliteitsresultaten waarmee de afstand tussen programma's kan worden vastgesteld en kan worden geverifieerd of een programma-specificatie aan een bepaalde compositionaliteitseigenschap voldoet. Complementair aan de resultaten over structurele specificatie ontwikkelen we een denotationeel model dat op een gestratifieerde manier het primitieve gedrag van programma's formaliseert en de compositionaliteitseigenschap van taal-operatoren bepaalt. Tenslotte levert een equationele axiomatisatie van de gedrags-afstand voor taaloperatoren een algoritmische aanpak om algebraïsche eigenschappen af te leiden zoals de afstand tussen programma's.

Titles in the IPA Dissertation Series since 2009

M.H.G. Verhoef. *Modeling and Validating Distributed Embedded Real-Time Control Systems*. Faculty of Science, Mathematics and Computer Science, RU. 2009-01

M. de Mol. *Reasoning about Functional Programs: Sparkle, a proof assistant for Clean*. Faculty of Science, Mathematics and Computer Science, RU. 2009-02

M. Lormans. *Managing Requirements Evolution*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-03

M.P.W.J. van Osch. *Automated Model-based Testing of Hybrid Systems*. Faculty of Mathematics and Computer Science, TU/e. 2009-04

H. Sozer. *Architecting Fault-Tolerant Software Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-05

M.J. van Weerdenburg. *Efficient Rewriting Techniques*. Faculty of Mathematics and Computer Science, TU/e. 2009-06

H.H. Hansen. *Coalgebraic Modelling: Applications in Automata Theory and Modal Logic*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2009-07

A. Mesbah. *Analysis and Testing of Ajax-based Single-page Web Applications*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-08

A.L. Rodriguez Yakushev. *Towards Getting Generic Programming Ready for Prime Time*. Faculty of Science, UU. 2009-9

K.R. Olmos Joffré. *Strategies for Context Sensitive Program Transformation*. Faculty of Science, UU. 2009-10

J.A.G.M. van den Berg. *Reasoning about Java programs in PVS using JML*. Faculty of Science, Mathematics and Computer Science, RU. 2009-11

M.G. Khatib. *MEMS-Based Storage Devices. Integration in Energy-Constrained Mobile Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-12

S.G.M. Cornelissen. *Evaluating Dynamic Analysis Techniques for Program Comprehension*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-13

D. Bolzoni. *Revisiting Anomaly-based Network Intrusion Detection Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-14

H.L. Jonker. *Security Matters: Privacy in Voting and Fairness in Digital Exchange*. Faculty of Mathematics and Computer Science, TU/e. 2009-15

M.R. Czenko. *TuLiP - Reshaping Trust Management*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-16

T. Chen. *Clocks, Dice and Processes*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2009-17

C. Kaliszyk. *Correctness and Availability: Building Computer Algebra on top of Proof Assistants and making Proof Assistants available over the Web*. Faculty of Science, Mathematics and Computer Science, RU. 2009-18

R.S.S. O'Connor. *Incompleteness & Completeness: Formalizing Logic and Analysis in Type Theory*. Faculty of Science, Mathematics and Computer Science, RU. 2009-19

B. Ploeger. *Improved Verification Methods for Concurrent Systems*. Faculty of Mathematics and Computer Science, TU/e. 2009-20

T. Han. *Diagnosis, Synthesis and Analysis of Probabilistic Models*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-21

R. Li. *Mixed-Integer Evolution Strategies for Parameter Optimization and Their Applications to Medical Image Analysis*. Faculty of Mathematics and Natural Sciences, UL. 2009-22

J.H.P. Kwisthout. *The Computational Complexity of Probabilistic Networks*. Faculty of Science, UU. 2009-23

T.K. Cocx. *Algorithmic Tools for Data-Oriented Law Enforcement*. Faculty of Mathematics and Natural Sciences, UL. 2009-24

A.I. Baars. *Embedded Compilers*. Faculty of Science, UU. 2009-25

M.A.C. Dekker. *Flexible Access Control for Dynamic Collaborative Environments*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-26

J.F.J. Laros. *Metrics and Visualisation for Crime Analysis and Genomics*. Faculty of Mathematics and Natural Sciences, UL. 2009-27

C.J. Boogerd. *Focusing Automatic Code Inspections*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2010-01

M.R. Neuhäußer. *Model Checking Nondeterministic and Randomly Timed Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2010-02

J. Endrullis. *Termination and Productivity*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2010-03

T. Staijen. *Graph-Based Specification and Verification for Aspect-Oriented Languages*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2010-04

Y. Wang. *Epistemic Modelling and Protocol Dynamics*. Faculty of Science, UvA. 2010-05

J.K. Berendsen. *Abstraction, Prices and Probability in Model Checking Timed Automata*. Faculty of Science, Mathematics and Computer Science, RU. 2010-06

A. Nugroho. *The Effects of UML Modeling on the Quality of Software*. Faculty of Mathematics and Natural Sciences, UL. 2010-07

A. Silva. *Kleene Coalgebra*. Faculty of Science, Mathematics and Computer Science, RU. 2010-08

J.S. de Bruin. *Service-Oriented Discovery of Knowledge - Foundations, Implementations and Applications*. Faculty of Mathematics and Natural Sciences, UL. 2010-09

D. Costa. *Formal Models for Component Connectors*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2010-10

M.M. Jaghoori. *Time at Your Service: Schedulability Analysis of Real-Time and Distributed Services*. Faculty of Mathematics and Natural Sciences, UL. 2010-11

R. Bakhshi. *Gossiping Models: Formal Analysis of Epidemic Protocols*. Faculty of Sciences, Department of Computer Science, VUA. 2011-01

B.J. Arnoldus. *An Illumination of the Template Enigma: Software Code Generation with Templates*. Faculty of Mathematics and Computer Science, TU/e. 2011-02

E. Zambon. *Towards Optimal IT Availability Planning: Methods and Tools*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-03

L. Astefanoaei. *An Executable Theory of Multi-Agent Systems Refinement*. Faculty of Mathematics and Natural Sciences, UL. 2011-04

J. Proença. *Synchronous coordination of distributed components*. Faculty of Mathematics and Natural Sciences, UL. 2011-05

A. Moralı. *IT Architecture-Based Confidentiality Risk Assessment in Networks of Organizations*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-06

M. van der Bijl. *On changing models in Model-Based Testing*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-07

C. Krause. *Reconfigurable Component Connectors*. Faculty of Mathematics and Natural Sciences, UL. 2011-08

M.E. Andrés. *Quantitative Analysis of Information Leakage in Probabilistic and Nondeterministic Systems*. Faculty of Science, Mathematics and Computer Science, RU. 2011-09

M. Atif. *Formal Modeling and Verification of Distributed Failure Detectors*. Faculty of Mathematics and Computer Science, TU/e. 2011-10

P.J.A. van Tilburg. *From Computability to Executability – A process-theoretic view on automata theory*. Faculty of Mathematics and Computer Science, TU/e. 2011-11

Z. Protic. *Configuration management for models: Generic methods for model comparison and model co-evolution*. Faculty of Mathematics and Computer Science, TU/e. 2011-12

S. Georgievska. *Probability and Hiding in Concurrent Processes*. Faculty of Mathematics and Computer Science, TU/e. 2011-13

S. Malakuti. *Event Composition Model: Achieving Naturalness in Runtime Enforcement*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-14

M. Raffelsieper. *Cell Libraries and Verification*. Faculty of Mathematics and Computer Science, TU/e. 2011-15

C.P. Tsirogiannis. *Analysis of Flow and Visibility on Triangulated Terrains*. Faculty of Mathematics and Computer Science, TU/e. 2011-16

Y.-J. Moon. *Stochastic Models for Quality of Service of Component Connectors*. Faculty of Mathematics and Natural Sciences, UL. 2011-17

R. Middelkoop. *Capturing and Exploiting Abstract Views of States in OO Verification*. Faculty of Mathematics and Computer Science, TU/e. 2011-18

M.F. van Amstel. *Assessing and Improving the Quality of Model Transformations*. Faculty of Mathematics and Computer Science, TU/e. 2011-19

A.N. Tamalet. *Towards Correct Programs in Practice*. Faculty of Science, Mathematics and Computer Science, RU. 2011-20

H.J.S. Basten. *Ambiguity Detection for Programming Language Grammars*. Faculty of Science, UvA. 2011-21

M. Izadi. *Model Checking of Component Connectors*. Faculty of Mathematics and Natural Sciences, UL. 2011-22

L.C.L. Kats. *Building Blocks for Language Workbenches*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2011-23

S. Kemper. *Modelling and Analysis of Real-Time Coordination Patterns*. Faculty of Mathematics and Natural Sciences, UL. 2011-24

J. Wang. *Spiking Neural P Systems*. Faculty of Mathematics and Natural Sciences, UL. 2011-25

A. Khosravi. *Optimal Geometric Data Structures*. Faculty of Mathematics and Computer Science, TU/e. 2012-01

A. Middelkoop. *Inference of Program Properties with Attribute Grammars, Revisited*. Faculty of Science, UU. 2012-02

Z. Hemel. *Methods and Techniques for the Design and Implementation of Domain-Specific Languages*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2012-03

T. Dimkov. *Alignment of Organizational Security Policies: Theory and Practice*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2012-04

S. Sedghi. *Towards Provably Secure Efficiently Searchable Encryption*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2012-05

F. Heidarian Dehkordi. *Studies on Verification of Wireless Sensor Networks and Abstraction Learning for System Inference*. Faculty of Science, Mathematics and Computer Science, RU. 2012-06

K. Verbeek. *Algorithms for Cartographic Visualization*. Faculty of Mathematics and Computer Science, TU/e. 2012-07

D.E. Nadales Agut. *A Compositional Interchange Format for Hybrid Systems: Design and Implementation*. Faculty of Mechanical Engineering, TU/e. 2012-08

H. Rahmani. *Analysis of Protein-Protein Interaction Networks by Means of Annotated Graph Mining Algorithms*. Faculty of Mathematics and Natural Sciences, UL. 2012-09

S.D. Vermolen. *Software Language Evolution*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2012-10

L.J.P. Engelen. *From Napkin Sketches to Reliable Software*. Faculty of Mathematics and Computer Science, TU/e. 2012-11

F.P.M. Stappers. *Bridging Formal Models – An Engineering Perspective*. Faculty of Mathematics and Computer Science, TU/e. 2012-12

W. Heijstek. *Software Architecture Design in Global and Model-Centric Software Development*. Faculty of Mathematics and Natural Sciences, UL. 2012-13

C. Kop. *Higher Order Termination*. Faculty of Sciences, Department of Computer Science, VUA. 2012-14

A. Osaiweran. *Formal Development of Control Software in the Medical Systems Domain*. Faculty of Mathematics and Computer Science, TU/e. 2012-15

W. Kuijper. *Compositional Synthesis of Safety Controllers*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2012-16

H. Beohar. *Refinement of Communication and States in Models of Embedded Systems*. Faculty of Mathematics and Computer Science, TU/e. 2013-01

G. Igna. *Performance Analysis of Real-Time Task Systems using Timed Automata*. Faculty of Science, Mathematics and Computer Science, RU. 2013-02

E. Zambon. *Abstract Graph Transformation – Theory and Practice*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2013-03

B. Lijnse. *TOP to the Rescue – Task-Oriented Programming for Incident Response Applications*. Faculty of Science, Mathematics and Computer Science, RU. 2013-04

G.T. de Koning Gans. *Outsmarting Smart Cards*. Faculty of Science, Mathematics and Computer Science, RU. 2013-05

M.S. Greiler. *Test Suite Comprehension for Modular and Dynamic Systems*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2013-06

L.E. Mamane. *Interactive mathematical documents: creation and presentation*. Faculty of Science, Mathematics and Computer Science, RU. 2013-07

M.M.H.P. van den Heuvel. *Composition and synchronization of real-time components upon one processor*. Faculty of Mathematics and Computer Science, TU/e. 2013-08

J. Businge. *Co-evolution of the Eclipse Framework and its Third-party Plug-ins*. Faculty of Mathematics and Computer Science, TU/e. 2013-09

S. van der Burg. *A Reference Architecture for Distributed Software Deployment*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2013-10

J.J.A. Keiren. *Advanced Reduction Techniques for Model Checking*. Faculty of Mathematics and Computer Science, TU/e. 2013-11

D.H.P. Gerrits. *Pushing and Pulling: Computing push plans for disk-shaped robots, and dynamic labelings for moving points*. Faculty of Mathematics and Computer Science, TU/e. 2013-12

M. Timmer. *Efficient Modelling, Generation and Analysis of Markov Automata*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2013-13

M.J.M. Roeloffzen. *Kinetic Data Structures in the Black-Box Model*. Faculty of Mathematics and Computer Science, TU/e. 2013-14

L. Lensink. *Applying Formal Methods in Software Development*. Faculty of Science, Mathematics and Computer Science, RU. 2013-15

C. Tankink. *Documentation and Formal Mathematics — Web Technology meets Proof Assistants*. Faculty of Science, Mathematics and Computer Science, RU. 2013-16

C. de Gouw. *Combining Monitoring with Run-time Assertion Checking*. Faculty of Mathematics and Natural Sciences, UL. 2013-17

J. van den Bos. *Gathering Evidence: Model-Driven Software Engineering in Automated Digital Forensics*. Faculty of Science, UvA. 2014-01

D. Hadziosmanovic. *The Process Matters: Cyber Security in Industrial Control Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-02

A.J.P. Jeckmans. *Cryptographically-Enhanced Privacy for Recommender Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-03

C.-P. Bezemer. *Performance Optimization of Multi-Tenant Software Systems*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2014-04

T.M. Ngo. *Qualitative and Quantitative Information Flow Analysis for Multi-threaded Programs*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-05

A.W. Laarman. *Scalable Multi-Core Model Checking*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-06

J. Winter. *Coalgebraic Characterizations of Automata-Theoretic Classes*. Faculty of Science, Mathematics and Computer Science, RU. 2014-07

W. Meulemans. *Similarity Measures and Algorithms for Cartographic Schematization*. Faculty of Mathematics and Computer Science, TU/e. 2014-08

A.F.E. Belinfante. *JTorX: Exploring Model-Based Testing*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-09

A.P. van der Meer. *Domain Specific Languages and their Type Systems*. Faculty of Mathematics and Computer Science, TU/e. 2014-10

B.N. Vasilescu. *Social Aspects of Collaboration in Online Software Communities*. Faculty of Mathematics and Computer Science, TU/e. 2014-11

F.D. Aarts. *Tomte: Bridging the Gap between Active Learning and Real-World Systems*. Faculty of Science, Mathematics and Computer Science, RU. 2014-12

N. Noroozi. *Improving Input-Output Conformance Testing Theories*. Faculty of Mathematics and Computer Science, TU/e. 2014-13

M. Helvensteijn. *Abstract Delta Modeling: Software Product Lines and Beyond*. Faculty of Mathematics and Natural Sciences, UL. 2014-14

P. Vullers. *Efficient Implementations of Attribute-based Credentials on Smart Cards*. Faculty of Science, Mathematics and Computer Science, RU. 2014-15

F.W. Takes. *Algorithms for Analyzing and Mining Real-World Graphs*. Faculty of Mathematics and Natural Sciences, UL. 2014-16

M.P. Schraagen. *Aspects of Record Linkage*. Faculty of Mathematics and Natural Sciences, UL. 2014-17

G. Alpár. *Attribute-Based Identity Management: Bridging the Cryptographic Design of ABCs with the Real World*. Faculty of Science, Mathematics and Computer Science, RU. 2015-01

A.J. van der Ploeg. *Efficient Abstractions for Visualization and Interaction*. Faculty of Science, UvA. 2015-02

R.J.M. Theunissen. *Supervisory Control in Health Care Systems*. Faculty of Mechanical Engineering, TU/e. 2015-03

T.V. Bui. *A Software Architecture for Body Area Sensor Networks: Flexibility and Trustworthiness*. Faculty of Mathematics and Computer Science, TU/e. 2015-04

A. Guzzi. *Supporting Developers' Teamwork from within the IDE*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2015-05

T. Espinha. *Web Service Growing Pains: Understanding Services and Their Clients*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2015-06

S. Dietzel. *Resilient In-network Aggregation for Vehicular Networks*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2015-07

E. Costante. *Privacy throughout the Data Cycle*. Faculty of Mathematics and Computer Science, TU/e. 2015-08

S. Cranen. *Getting the point — Obtaining and understanding fixpoints in model checking*. Faculty of Mathematics and Computer Science, TU/e. 2015-09

R. Verdult. *The (in)security of proprietary cryptography*. Faculty of Science, Mathematics and Computer Science, RU. 2015-10

J.E.J. de Ruiter. *Lessons learned in the analysis of the EMV and TLS security protocols*. Faculty of Science, Mathematics and Computer Science, RU. 2015-11

Y. Dajsuren. *On the Design of an Architecture Framework and Quality Evaluation for Automotive Software Systems*. Faculty of Mathematics and Computer Science, TU/e. 2015-12

J. Bransen. *On the Incremental Evaluation of Higher-Order Attribute Grammars*. Faculty of Science, UU. 2015-13

S. Picek. *Applications of Evolutionary Computation to Cryptology*. Faculty of Science, Mathematics and Computer Science, RU. 2015-14

C. Chen. *Automated Fault Localization for Service-Oriented Software Systems*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2015-15

S. te Brinke. *Developing Energy-Aware Software*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2015-16

R.W.J. Kersten. *Software Analysis Methods for Resource-Sensitive Systems*. Faculty of Science, Mathematics and Computer Science, RU. 2015-17

J.C. Rot. *Enhanced coinduction*. Faculty of Mathematics and Natural Sciences, UL. 2015-18

M. Stolikj. *Building Blocks for the Internet of Things*. Faculty of Mathematics and Computer Science, TU/e. 2015-19

D. Gebler. *Robust SOS Specifications of Probabilistic Processes*. Faculty of Sciences, Department of Computer Science, VUA. 2015-20