The Power of Convex Algebras

Filippo Bonchi[1], Alexandra Silva[2], and Ana Sokolova[3]

1 **ENS de Lyon, France** — `filippo.bonchi@ens-lyon.fr`
2 **UCL, UK** — `alexandra.silva@ucl.ac.uk`
3 **University of Salzburg, Austria** — `ana.sokolova@cs.uni-salzburg.at`

Abstract

Probabilistic automata (PA) combine probability and nondeterminism. They can be given different semantics, like strong bisimilarity, convex bisimilarity, or (more recently) distribution bisimilarity. The latter is based on the view of PA as transformers of probability distributions, also called belief states, and promotes distributions to first-class citizens.

We give a coalgebraic account of the latter semantics, and explain the genesis of the belief-state transformer from a PA. To do so, we make explicit the convex algebraic structure present in PA and identify belief-state transformers as transition systems with state space that carries a convex algebra. As a consequence of our abstract approach, we can give a sound proof technique which we call bisimulation up-to convex hull.

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Probabilistic automata (PA), closely related to Markov decision processes (MDPs), have been used along the years in various areas of verification [40, 37, 38, 2], machine learning [24, 41], and semantics [66, 51]. Recent interest in research around semantics of probabilistic programming languages has led to new insights in connections between category theory, probability theory, and automata [58, 12, 27, 57, 44].

PA have been given various semantics, starting from strong bisimilarity [39], probabilistic (convex) bisimilarity [49, 48], to bisimilarity on distributions [18, 13, 10, 21, 11, 25, 22, 26]. In this last view, probabilistic automata are understood as transformers of belief states, labeled transition systems (LTSs) having as states probability distributions, see e.g. [13, 15, 35, 1, 14, 22, 19]. Checking such equivalence raises a lot of challenges since belief-states are uncountable. Nevertheless, it is decidable [26, 20] with help of convexity. Despite these developments, what remains open is the understanding of the genesis of belief-state transformers and canonicity of distribution bisimilarity, as well as the role of convex algebras.

The theory of coalgebras [30, 46, 33] provides a tool-box for modelling and analysing different types of state machines. In a nutshell, a coalgebra is an arrow $c\colon S \to FS$ for some functor $F\colon \mathbf{C} \to \mathbf{C}$ on a category $\mathbf{C}$. Intuitively S represents the space of states of the machine, c its transition structure and the functor F its type. Most importantly, every functor gives rise to a canonical notion of behavioural equivalence ($\approx$), a coinductive proof technique and, for finite states machines, a procedure to check $\approx$.

By tuning the parameters $\mathbf{C}$ and F, one can retrieve many existing types of machines and their associated equivalences. For instance, by taking $\mathbf{C} = \mathbf{Sets}$, the category of sets and functions, and $FS = (\mathcal{P}\mathcal{D}S)^L$, the set of functions from L to subsets ($\mathcal{P}$) of probability distributions ($\mathcal{D}$) over S, coalgebras $c\colon S \to FS$ are in one-to-one correspondence with PA with labels in L. Moreover, the associated notion of behavioural equivalence turns out to be the classical strong probabilistic bisimilarity of [39] (see [4, 53] for more details). Recent work [43] shows that, by taking a slightly different functor, forcing the subsets to be convex, one obtains probabilistic (convex) bisimilarity as in [49, 48].

In this paper, we take a coalgebraic outlook at the semantics of probabilistic automata as belief-state transformers: we wish to translate a PA $c\colon S \to (\mathcal{PD}S)^L$ into a belief state transformer $c^\sharp\colon \mathcal{D}S \to (\mathcal{PD}S)^L$. Note that the latter is a coalgebra for the functor $FX = (\mathcal{P}X)^L$, i.e., a labeled transition system, since the state space is the set of probability distributions $\mathcal{D}S$. This is reminiscent of the standard determinisation for non-deterministic automata (NDA) seen as coalgebras $c\colon S \to 2 \times (\mathcal{P}S)^A$. The result of the determinisation is a deterministic automaton $c^\sharp\colon \mathcal{P}S \to 2 \times (\mathcal{P}S)^A$ (with state space $\mathcal{P}S$), which is a coalgebra for the functor $FX = 2 \times X^A$. In the case of PA, one lifts the states space to $\mathcal{D}S$, in the one of NDA to $\mathcal{P}S$. From an abstract perspective, both $\mathcal{D}$ and $\mathcal{P}$ are monads, hereafter denoted by $\mathcal{M}$, and both PA and NDA can be regarded as coalgebras of type $c\colon S \to F\mathcal{M}S$.

In [52], a generalised determinisation transforming coalgebras $c\colon S \to F\mathcal{M}S$ into coalgebras $c^\sharp\colon \mathcal{M}S \to F\mathcal{M}S$ was presented. This construction requires the existence of a *lifting* $\overline{F}$ of F to the category of algebras for the monad $\mathcal{M}$. In the case of NDA, the functor $FX = 2 \times X^A$ can be easily lifted to the category of join-semilattices (algebras for $\mathcal{P}$) and, the coalgebra $c^\sharp\colon \mathcal{P}S \to 2 \times (\mathcal{P}S)^A$ resulting from this construction turns out to be exactly the standard determinised automaton. Unfortunately, this is not the case with probabilistic automata: because of the lack of a suitable distributive law of $\mathcal{D}$ over $\mathcal{P}$ [63], it is impossible to suitably lift $FX = (\mathcal{P}X)^L$ to the category of *convex algebras* (algebras for the monad $\mathcal{D}$).

The way out of the impasse consists in defining a powerset-like functor on the category of convex algebras. This is not a lifting but it enjoys enough properties that allow to lift every PA into a labeled transition system on convex algebras. In turn, these can be transformed – without changing the underlying behavioural equivalence – into standard LTSs on **Sets** by simply forgetting the algebraic structure. We show that the result of the whole procedure is exactly the expected belief-state transformer and that the induced notion of behavioural equivalence coincides with a canonical one present in the literature [13, 25, 22, 26].

The analogy with NDA pays back in terms of proof techniques. In [6], Bonchi and Pous introduced an efficient algorithm to check language equivalence of NDA based on coinduction up-to [45]: in a determinised automaton $c^\sharp\colon \mathcal{P}S \to 2 \times (\mathcal{P}S)^A$, language equivalence can be proved by means of bisimulations up-to the structure of join semilattice carried by the state space $\mathcal{P}S$. Algorithmically, this results in an impressive pruning of the search space.

Similarly, in a belief-state transformer $c^\sharp\colon \mathcal{D}S \to (\mathcal{PD}S)^L$, one can coinductively reason up-to the convex algebraic structure carried by $\mathcal{D}S$. The resulting proof technique, which we call in this paper *bisimulation up-to convex hull*, allows finite relations to witness the equivalence of infinitely many states. More precisely, by exploiting a recent result in convex algebra by Sokolova and Woracek [55], we are able to show that the equivalence of any two belief states can always be proven by means of a *finite* bisimulation up-to.

The paper starts with background on PA (Section 2), convex algebras (Section 3), and coalgebra (Section 4). We provide the PA functor on convex algebras in Section 5. We give the transformation from PA to belief-state transformers in Section 6 and prove the coincidence of the abstract and concrete transformers and semantics. We present bisimulation up-to convex hull and prove soundness in Section 7. Proofs of all results are in appendix.

2 Probabilistic Automata

Probabilistic automata are models of systems that involve both probability and nondeterminism. We start with their definition by Segala and Lynch [49].

▶ **Definition 1.** A probabilistic automaton (PA) is a triple $M = (S, L, \to)$ where S is a set of states, L is a set of actions or action labels, and $\to\, \subseteq\; S \times L \times \mathcal{D}(S)$ is the transition

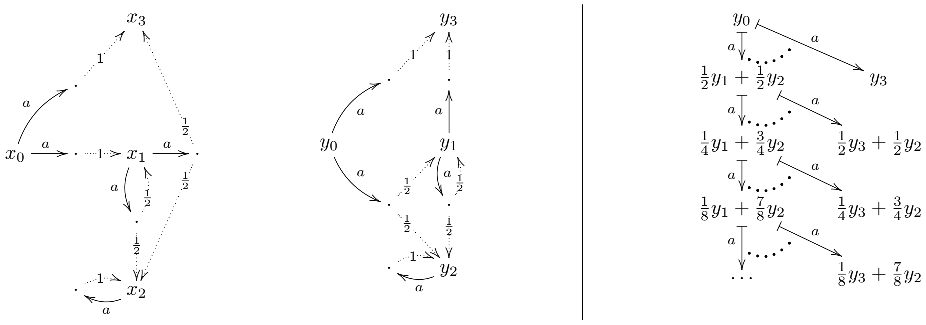

Figure 1 On the left: a PA with set of actions $L = \{a\}$ and set of states $S = \{x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4\}$. We depict each transition $s \xrightarrow{a} \zeta$ in two stages: a straight action-labeled arrow from s to $\cdot$ and then several dotted arrows from $\cdot$ to states in S specifing the distribution ζ. On the right: part of the corresponding belief-state transformer. The dots between two arrows $\zeta \overset{a}{\mapsto} \xi_1$ and $\zeta \overset{a}{\mapsto} \xi_2$ denote that ζ can perform infinitely many transitions to states obtained as convex combinations of ξ_1 and ξ_2. For instance $y_0 \overset{a}{\mapsto} \frac{1}{4}y_1 + \frac{1}{4}y_2 + \frac{1}{2}y_3$.

relation. As usual, $s \xrightarrow{a} \zeta$ stands for $(s, a, \zeta) \in \to$. ◇

An example is shown on the left of Figure 1. Probabilistic automata can be given different semantics, e.g., (strong probabilistic) bisimilarity [39], convex (probabilistic) bisimilarity [49], and as transformers of belief states [10, 22, 14, 15, 13, 26] whose definitions we present next. For the rest of the section, we fix a PA $M = (S, L, \to)$.

▶ **Definition 2** (Strong Probabilistic Bisimilarity). A relation $R \subseteq S \times S$ is a (strong probabilistic) *bisimulation* if $(s,t) \in R$ implies, for all actions $a \in L$ and all $\xi \in \mathcal{D}(S)$, that

$$s \xrightarrow{a} \xi \Rightarrow \exists \xi' \in \mathcal{D}(S).\ t \xrightarrow{a} \xi' \wedge \xi \equiv_R \xi', \quad \text{and} \quad t \xrightarrow{a} \xi' \Rightarrow \exists \xi \in \mathcal{D}(S).\ s \xrightarrow{a} \xi \wedge \xi \equiv_R \xi'.$$

Here, $\equiv_R \subseteq \mathcal{D}(S) \times \mathcal{D}(S)$ is the lifting of R to distributions, defined by $\xi \equiv_R \xi'$ if and only if there exists a distribution $\nu \in \mathcal{D}(S \times S)$ such that
1. $\sum_{t \in S} \nu(s,t) = \xi(s)$ for any $s \in S$, 2. $\sum_{s \in S} \nu(s,t) = \xi'(t)$ for any $t \in T$, and
3. $\nu(s,t) \neq 0$ implies $(s,t) \in R$.

Two states s and t are (strongly probabilistically) *bisimilar*, notation $s \sim t$, if there exists a (strong probabilistic) bisimulation R with $(s,t) \in R$. ◇

▶ **Definition 3** (Convex Bisimilarity). A relation $R \subseteq S \times S$ is a *convex* (probabilistic) *bisimulation* if $(s,t) \in R$ implies, for all actions $a \in L$ and all $\xi \in \mathcal{D}(S)$, that

$$s \xrightarrow{a} \xi \Rightarrow \exists \xi' \in \mathcal{D}(S).\ t \xrightarrow{a}_c \xi' \wedge \xi \equiv_R \xi', \quad \text{and} \quad t \xrightarrow{a} \xi' \Rightarrow \exists \xi \in \mathcal{D}(S).\ s \xrightarrow{a}_c \xi \wedge \xi \equiv_R \xi'.$$

Here $\to_c$ denotes the convex transition relation, defined as follows: $s \xrightarrow{a}_c \xi$ if and only if $\xi = \sum_{i=1}^n p_i \xi_i$ for some $\xi_i \in \mathcal{D}(S)$ and $p_i \in [0,1]$ satisfying $\sum_{i=1}^n p_i = 1$ and $s \xrightarrow{a} \xi_i$ for $i = 1, \ldots, n$. Two states s and t are *convex bisimilar*, notation $s \sim_c t$, if there exists a convex bisimulation R with $(s,t) \in R$. ◇

Convex bisimilarity is (strong probabilistic) bisimilarity on the "convex closure" of the given PA. More precisely, consider the PA $M_c = (S, L, \to_c)$ in which $s \xrightarrow{a}_c \xi$ whenever $s \in S$ and ξ is in the convex hull (see Section 3 for a definition) of the set $\{\zeta \in \mathcal{D}(S) \mid s \xrightarrow{a} \zeta\}$. Then convex bisimilarity of M is bisimilarity of M_c. Hence, if bisimilarity is the behavioural equivalence of interest, we see that convex semantics arises from a different perspective on the representation of a PA: instead of seeing the given transitions as independent, we look at them as generators of infinitely many transitions in the convex closure.

There is yet another way to understand PA, as belief-state transformers, present but sometimes implicit in [10, 25, 22, 14, 15, 13, 26, 11] to name a few, with behavioural equivalences on distributions. We were particularly inspired by the original work of Deng et al. [14, 15, 13] as well as [26]. Given a PA $M = (S, L, \to)$, consider the labeled transition system $M_{bs} = (\mathcal{D}S, L, \mapsto)$ with states distributions over the original states of M, and transitions $\mapsto \subseteq \mathcal{D}S \times L \times \mathcal{D}S$ defined by

$$\xi \overset{a}{\mapsto} \zeta \quad \text{iff} \quad \xi = \sum p_i s_i,\ s_i \xrightarrow{a}_c \xi_i,\ \zeta = \sum p_i \xi_i.$$

We call M_{bs} the belief-state transformer of M. Figure 1, right, displays a part of the belief-state transformer induced by the PA of Figure 1, left. According to this definition, a distribution makes an action step only if all its support states can make the step.

This, and hence the corresponding notion of bisimulation, can vary. For example, in [26] a distribution makes a transition $\overset{a}{\mapsto}$ if some of its support states can perform an $\overset{a}{\mapsto}$ step[1]. There are several proposed notions of equivalences on distributions [25, 18, 19, 22, 14, 10, 26] that mainly differ in the treatment of termination[2]. See [26] for an extensive survey of related work.

▶ **Definition 4** (Distribution Bisimilarity)**.** An equivalence $R \subseteq \mathcal{D}S \times \mathcal{D}S$ is a distribution bisimulation of M if and only if it is a bisimulation of the belief-state transformer M_{bs}.

Two distributions ξ and ζ are *distribution bisimilar*, notation $\xi \sim_d \zeta$, if there exists a bisimulation R with $(\xi, \zeta) \in R$. Two states s and t are *distribution bisimilar*, notation $s \sim_d t$, if $\delta_s \sim_d \delta_t$, where δ_x denotes the Dirac distribution with $\delta_x(x) = 1$. ◇

While the foundations of strong probabilistic bisimilarity are well-studied [53, 4, 65] and convex probabilistic bisimilarity was also recently captured coalgebraically [43], the foundations of the semantics of PA as transformers of belief states is not yet explained. One of the goals of the present paper is to show that also that semantics (naturally on distributions [26]) is an instance of generic behavioural equivalence. Note that a (somewhat concrete) proof is given for the bisimilarity of [26] — the authors have proven that their bisimilarity is coalgebraic bisimilarity of a certain coalgebra corresponding to the belief-state transformer. What is missing there, and in all related work, is an explanation of the relationship of the belief-state transformer to the the original PA. Clarifying the foundations of the belief-state transformer and the distribution bisimilarity is our initial motivation.

3 Convex Algebras

By $\mathcal{C}$ we denote the signature of convex algebras

$$\mathcal{C} = \{(p_i)_{i=0}^n \mid n \in \mathbb{N}, p_i \in [0,1], \sum_{i=0}^n p_i = 1\}.$$

The operation symbol $(p_i)_{i=0}^n$ has arity $(n+1)$ and it will be interpreted by a convex combination with coefficients p_i for $i = 0, \ldots, n$. For a real number $p \in [0,1]$ we set $\bar{p} = 1 - p$.

[1] While the definition of [26] is practically powerful and interesting exactly for its special treatment of termination, and it was a valuable source of motivation for our work, it is not the most canonical when it comes to providing coalgebraic foundation.

[2] There are also some notions of equivalences on distributions that amount to strong / convex bisimilarity [25, 11] and are not interesting for our work.

▶ **Definition 5.** A *convex algebra* $\mathbb{X}$ is an algebra with signature $\mathcal{C}$, i.e., a set X together with an operation $\sum_{i=0}^{n} p_i(-)_i$ for each operational symbol $(p_i)_{i=0}^{n} \in \mathcal{C}$, such that the following two axioms hold:

- Projection: $\sum_{i=0}^{n} p_i x_i = x_j$ if $p_j = 1$.
- Barycenter: $\sum_{i=0}^{n} p_i \left(\sum_{j=0}^{m} q_{i,j} x_j\right) = \sum_{j=0}^{m} \left(\sum_{i=0}^{n} p_i q_{i,j}\right) x_j$.

A convex algebra homomorphism h from $\mathbb{X}$ to $\mathbb{Y}$ is a *convex* (synonymously, *affine*) map, i.e., $h\colon X \to Y$ with the property $h\left(\sum_{i=0}^{n} p_i x_i\right) = \sum_{i=0}^{n} p_i h(x_i)$. ◇

▶ Remark 6. Let $\mathbb{X}$ be a convex algebra. Then (for $p_n \neq 1$)

$$\sum_{i=0}^{n} p_i x_i = \overline{p_n} \left(\sum_{j=0}^{n-1} \frac{p_j}{\overline{p_n}} x_j \right) + p_n x_n \tag{1}$$

Hence, an $(n+1)$-ary convex combination can be written as a binary convex combination using an n-ary convex combination. As a consequence, if X is a set that carries two convex algebras $\mathbb{X}_1$ and $\mathbb{X}_2$ with operations $\sum_{i=0}^{n} p_i(-)_i$ and $\bigoplus_{i=0}^{n} p_i(-)_i$, respectively (and binary versions $+$ and $\oplus$, respectively) such that $px + \bar{p}y = px \oplus \bar{p}y$ for all p, x, y, then $\mathbb{X}_1 = \mathbb{X}_2$.

One can also see (1) as a definition, see e.g. [59, Definition 1]. We make the connection explicit with the next proposition, cf. [59, Lemma 1-Lemma 4][3].

▶ **Proposition 7.** *Let X be a set with binary operations $px + \bar{p}y$ for $x, y \in X$ and $p \in (0,1)$. For $x, y, z \in X$ and $p, q \in (0,1)$, assume*

- *Idempotence: $px + \bar{p}x = x$,*
- *Parametric commutativity: $px + \bar{p}y = \bar{p}y + px$,*
- *Parametric associativity: $p(qx + \bar{q}y) + \bar{p}z = pqx + \overline{pq}\left(\frac{p\bar{q}}{\overline{pq}}y + \frac{\bar{p}}{\overline{pq}}z\right)$,*

and define n-ary convex operations by the projection axiom and the formula (1). Then X becomes a convex algebra. ◀

Hence, it suffices to consider binary convex combinations only, whenever more convenient.

▶ **Definition 8.** Let $\mathbb{X}$ be a convex algebra, with carrier X and $C \subseteq X$. C is *convex* if it is the carrier of a subalgebra of $\mathbb{X}$, i.e., if $px + \bar{p}y \in C$ for all $x, y \in C$ and $p \in (0,1)$. By $\mathrm{conv}(S)$ we denote the *convex hull* of a set $S \subseteq X$, i.e., $\mathrm{conv}(S)$ is the smallest convex set that contains S. ◇

Clearly, a set $C \subseteq X$ for X being the carrier of a convex algebra $\mathbb{X}$ is convex if and only if $C = \mathrm{conv}(C)$. Convexity plays an important role in the semantics of probabilistic automata, for example in the definition of convex bisimulation, Definition 3.

4 Coalgebras

In this section, we briefly review some notions from (co)algebra which we will use in the rest of the paper. This section is written for a reader familiar with basic category theory. We have included an expanded version of this section in Appendix D that also includes basic categorical definitions and more details than what we do here.

Coalgebras provide an abstract framework for state-based systems. Let $\mathbf{C}$ be a base category. A coalgebra is a pair (S, c) of a state space S (object in $\mathbf{C}$) and an arrow $c\colon S \to FS$ in $\mathbf{C}$ where $F\colon \mathbf{C} \to \mathbf{C}$ is a functor that specifies the type of transitions. We will sometimes

[3] Stone's cancellation Postulate V is not used in his Lemma 1-Lemma 4.

just say the coalgebra $c\colon S \to FS$, meaning the coalgebra (S, c). A coalgebra homomorphism from a coalgebra (S, c) to a coalgebra (T, d) is an arrow $h\colon S \to T$ in $\mathbf{C}$ that makes the diagram on the right commute. Coalgebras of a functor F and their coalgebra homomorphisms form a category that we denote by $\mathrm{Coalg}_{\mathbf{C}}(F)$. Examples of functors on **Sets** which are of interest to us are:

$$\begin{tikzcd} S \arrow[r, "h"] \arrow[d, "c"] & T \arrow[d, "d"] \\ FS \arrow[r, "Fh"] & FT \end{tikzcd}$$

1. The constant exponent functor $(-)^L$ for a set L, mapping a set X to the set X^L of all functions from L to X, and a function $f\colon X \to Y$ to $f^L\colon X^L \to Y^L$ with $f^A(g) = f \circ g$.
2. The powerset functor $\mathcal{P}$ mapping a set X to its powerset $\mathcal{P}X = \{S \mid S \subseteq X\}$ and on functions $f\colon X \to Y$ given by direct image: $\mathcal{P}f\colon \mathcal{P}X \to \mathcal{P}Y$, $\mathcal{P}(f)(U) = \{f(u) \mid u \in U\}$.
3. The finitely supported probability distribution functor $\mathcal{D}$ is defined, for a set X and a function $f\colon X \to Y$, as

$$\mathcal{D}X \;=\; \{\varphi\colon X \to [0,1] \mid \sum_{x \in X} \varphi(x) = 1, \operatorname{supp}(\varphi) \text{ is finite}\} \qquad \mathcal{D}f(\varphi)(y) \;=\; \sum_{x \in f^{-1}(y)} \varphi(x).$$

   The support set of a distribution $\varphi \in \mathcal{D}X$ is defined as $\operatorname{supp}(\varphi) = \{x \in X \mid \varphi(x) \neq 0\}$.
4. The functor C [43, 29, 62] maps a set X to the set of all nonempty convex subsets of distributions over X, and a function $f\colon X \to Y$ to the function $\mathcal{P}\mathcal{D}f$.

We will often decompose $\mathcal{P}$ as $\mathcal{P}_{ne} + 1$ where $\mathcal{P}_{ne}$ is the nonempty powerset functor and $(-) + 1$ is the termination functor defined for every set X by $X + 1 = X \cup *$ with $* \notin X$ and every function $f\colon X \to Y$ by $f + 1(*) = *$ and $f + 1(x) = x$ for $x \in X$.

Coalgebras over a concrete category are equipped with a generic behavioural equivalence, which we define next. Let (S, c) be an F-coalgebra on a concrete category $\mathbf{C}$, with $\mathcal{U}\colon \mathbf{C} \to \mathbf{Sets}$ being the forgetful functor. An equivalence relation $R \subseteq \mathcal{U}S \times \mathcal{U}S$ is a kernel bisimulation (synonymously, a cocongruence) [56, 36, 67] if it is the kernel of a homomorphism, i.e., $R = \ker \mathcal{U}h = \{(s,t) \in \mathcal{U}S \times \mathcal{U}S \mid \mathcal{U}h(s) = \mathcal{U}h(t)\}$ for some coalgebra homomorphism $h\colon (S, c) \to (T, d)$ to some F-coalgebra (T, d). Two states s, t of a coalgebra are behaviourally equivalent notation $s \approx t$ iff there is a kernel bisimulation R with $(s, t) \in R$. A trivial but important property is that if there is a functor from one category of coalgebras (over a concrete category) to another, then this functor preserves behavioural equivalence: if two states are equivalent in a coalgebra of the first category, then they are also equivalent in the image under the functor in the second category.

We are now in position to connect probabilistic automata to coalgebras.

▶ **Proposition 9** ([4, 53])**.** *A probabilistic automaton $M = (S, L, \to)$ can be identified with a $(\mathcal{P}\mathcal{D})^L$-coalgebra $c_M\colon S \to (\mathcal{P}\mathcal{D}S)^L$ on* **Sets***, where $s \xrightarrow{a} \xi$ in M iff $\xi \in c_M(s)(a)$ in (S, c_M). Bisimilarity in M equals behavioural equivalence in (S, c_M), i.e., for two states $s, t \in S$ we have $s \sim t \Leftrightarrow s \approx t$.* ◀

It is also possible to provide convex bisimilarity semantics to probabilistic automata via coalgebraic behavioural equivalence, as the next proposition shows.

▶ **Proposition 10** ([43])**.** *Let $M = (S, L, \to)$ be a probabilistic automaton, and let $(S, \bar{c}_M)$ be a $(C + 1)^L$-coalgebra on* **Sets** *defined by $\bar{c}_M(s)(a) = \operatorname{conv}(c_M(s)(a))$ where c_M is as before, if $c_M(s)(a) = \{\xi \mid s \xrightarrow{a} \xi\} \neq \emptyset$; and $\bar{c}_M(s)(a) = *$ if $c_M(s)(a) = \emptyset$. Convex bisimilarity in M equals behavioural equivalence in $(S, \bar{c}_M)$.* ◀

The connection between (S, c_M) and $(S, \bar{c}_M)$ in Proposition 10 is the same as the connection between M and M_c in Section 2. Abstractly, it can be explained using the following well known generic property.

▶ **Lemma 11** ([46, 4]). *Let $\sigma\colon F \Rightarrow G$ be a natural transformation from $F\colon \mathbf{C} \to \mathbf{C}$ to $G\colon \mathbf{C} \to \mathbf{C}$. Then $\mathcal{T}\colon \mathrm{Coalg}_{\mathbf{C}}(F) \to \mathrm{Coalg}_{\mathbf{C}}(G)$ given by $\mathcal{T}(S \xrightarrow{c} FS) = (S \xrightarrow{c} FS \xrightarrow{\sigma_S} GS)$ on objects and identity on morphisms is a functor. As a consequence, $\mathcal{T}$ preserves behavioural equivalence. If σ is injective, then $\mathcal{T}$ also reflects behavioural equivalence.* ◀

▶ **Example 12.** We have that $\mathrm{conv}\colon \mathcal{PD} \Rightarrow C + 1$ given by $\mathrm{conv}(\emptyset) = *$ and $\mathrm{conv}(X)$ is the already-introduced convex hull for $X \subseteq \mathcal{D}S$, $X \neq \emptyset$ is a natural transformation. Therefore, $\mathrm{conv}^L\colon (\mathcal{PD})^L \Rightarrow (C+1)^L$ is one as well, defined pointwise. As a consequence from Lemma 11, we get a functor $\mathcal{T}_{\mathrm{conv}}\colon \mathrm{Coalg}_{\mathbf{Sets}}((\mathcal{PD})^L) \to \mathrm{Coalg}_{\mathbf{Sets}}((C+1)^L)$ and hence bisimilarity implies convex bisimilarity in probabilistic automata.

On the other hand, we have the injective natural transformation $\iota\colon C + 1 \Rightarrow \mathcal{PD}$ given by $\iota(X) = X$ and $\iota(*) = \emptyset$ and hence a natural transformation $\chi\colon (C+1)^L \Rightarrow (\mathcal{PD})^L$. As a consequence, convex bisimilarity coincides with strong bisimilarity on the "convex-closed" probabilistic automaton M_c, i.e., the coalgebra $(S, \bar{c}_M)$ whose transitions are all convex combinations of M-transitions.

4.1 Algebras for a Monad

The behaviour functor F often is, or involves, a monad $\mathcal{M}$, providing certain computational effects, such as partial, non-deterministic, or probabilistic computations.

More precisely, a monad is a functor $\mathcal{M}\colon \mathbf{C} \to \mathbf{C}$ together with two natural transformations: a unit $\eta\colon \mathrm{id}_{\mathbf{C}} \Rightarrow \mathcal{M}$ and multiplication $\mu\colon \mathcal{M}^2 \Rightarrow \mathcal{M}$ that satisfy the laws $\mu \circ \eta_{\mathcal{M}} = id = \mu \circ \mathcal{M}\eta$ and $\mu \circ \mu_{\mathcal{M}} = \mu \circ \mathcal{M}\mu$. An example that will be pivotal for our exposition is the distribution monad.

- The unit of $\mathcal{D}$ is given by a Dirac distribution $\eta(x) = \delta_x = (x \mapsto 1)$ for $x \in X$ and the multiplication by $\mu(\Phi)(x) = \sum\limits_{\varphi \in \mathrm{supp}(\Phi)} \Phi(\varphi) \cdot \varphi(x)$ for $\Phi \in \mathcal{DD}X$.

With a monad $\mathcal{M}$ on a category $\mathbf{C}$ one associates the Eilenberg-Moore category $\mathrm{EM}(\mathcal{M})$ of Eilenberg-Moore algebras. Objects of $\mathrm{EM}(\mathcal{M})$ are pairs $\mathbb{A} = (A, a)$ of an object $A \in \mathbf{C}$ and an arrow $a\colon \mathcal{M}A \to A$, satisfying $a \circ \eta = id$ and $a \circ \mathcal{M}a = a \circ \mu$. A homomorphism from an algebra $\mathbb{A} = (A, a)$ to an algebra $\mathbb{B} = (B, b)$ is a map $h\colon A \to B$ in $\mathbf{C}$ between the underlying objects satisfying $h \circ a = b \circ \mathcal{M}h$.

A category of Eilenberg-Moore algebras which is particularly relevant for our exposition is described in the following proposition. See [60] and [50] for the original result, but also [16, 17] or [28, Theorem 4] where a concrete and simple proof is given.

▶ **Proposition 13** ([60, 16, 17, 28]). *Eilenberg-Moore algebras of the finitely supported distribution monad $\mathcal{D}$ are exactly convex algebras as defined in Section 3. The arrows in the Eilenberg-Moore category $\mathrm{EM}(\mathcal{D})$ are convex algebra homomorphisms.* ◀

As a consequence, we will interchangeably use the abstract (Eilenberg-Moore algebra) and the concrete definition (convex algebra), whatever is more convenient. For the latter, we also just use binary convex operations, by Proposition 7, whenever more convenient.

4.2 The Generalised Determinisation

We now recall a construction from [52], which serves as source of inspiration for our work.

A functor $\overline{F}\colon \mathrm{EM}(\mathcal{M}) \to \mathrm{EM}(\mathcal{M})$ is said to be a lifting of a functor $F\colon \mathbf{C} \to \mathbf{C}$ if and only if $\mathcal{U} \circ \overline{F} = F \circ \mathcal{U}$. Here, $\mathcal{U}$ is the forgetful functor $\mathcal{U}\colon \mathrm{EM}(\mathcal{M}) \to \mathbf{C}$ mapping an algebra to its carrier. It has a left adjoint $\mathcal{F}$, mapping an object $X \in \mathbf{C}$ to the (free) algebra $(\mathcal{M}X, \mu_X)$. We have that $\mathcal{M} = \mathcal{U} \circ \mathcal{F}$.

Whenever $F\colon \mathbf{C} \to \mathbf{C}$ has a lifting $\overline{F}\colon \ \mathrm{EM}(\mathcal{M}) \to \mathrm{EM}(\mathcal{M})$, one has the following functors between categories of coalgebras.

$$\begin{tikzcd} & \mathrm{Coalg}_{\mathrm{EM}(\mathcal{M})}\,(\overline{F}) \arrow[dr, "\overline{\mathcal{U}}"] & \\ \mathrm{Coalg}_{\mathbf{C}}\,(F\mathcal{M}) \arrow[ur, "\overline{\mathcal{F}}"] & & \mathrm{Coalg}_{\mathbf{C}}\,(F) \end{tikzcd}$$

The functor $\overline{\mathcal{F}}$ transforms every coalgebra $c\colon S \to F\mathcal{M}S$ over the base category into a coalgebra $c^{\sharp}\colon \mathcal{F}S \to \overline{F}\mathcal{F}S$. Note that this is a coalgebra on $\mathrm{EM}(\mathcal{M})$: the state space carries an algebra, actually the freely generated one, and $c^{\sharp}$ is a homomorphism of $\mathcal{M}$-algebras. Intuitively, this amounts to compositionality: like in GSOS specifications, the transitions of a compound state are determined by the transitions of its components.

The functor $\overline{\mathcal{U}}$ simply forgets about the algebraic structure: $c^{\sharp}$ is mapped into

$$\mathcal{U}c^{\sharp}\colon \mathcal{M}S = \mathcal{U}\mathcal{F}S \to \mathcal{U}\overline{F}\mathcal{F}S = F\mathcal{U}\mathcal{F}S = F\mathcal{M}S.$$

An important property of $\overline{\mathcal{U}}$ is that it preserves and reflects behavioural equivalence. On the one hand, this fact usually allows to give concrete characterisation of $\approx$ for $\overline{F}$-coalgebras. On the other, it allows, by means of the so-called up-to techniques, to exploit the $\mathcal{M}$-algebraic structure of $\mathcal{F}S$ to check $\approx$ on $\mathcal{U}c^{\sharp}$.

By taking $F = 2 \times (-)^L$ and $\mathcal{M} = \mathcal{P}$, one transforms $c\colon S \to 2 \times (\mathcal{P}S)^L$ into $\mathcal{U}c^{\sharp}\colon \mathcal{P}S \to 2 \times (\mathcal{P}S)^L$. The former is a non-deterministic automaton (every c of this type is a pairing $\langle o, t\rangle$ of $o\colon S \to 2$, defining the final states, and $t\colon S \to \mathcal{P}(S)^L$, defining the transition relation) and the latter is a deterministic automaton which has $\mathcal{P}S$ as states space. In [52], see also [31], it is shown that, for a certain choice of the lifting $\overline{F}$, this amounts exactly to the standard determinisation from automata theory. This explains why this construction is called *the generalised determinisation*.

In a sense, this is similar to the translation of probabilistic automata into belief-state transformers that we have seen in Section 2. Indeed, probabilistic automata are coalgebras $c\colon S \to (\mathcal{P}\mathcal{D}S)^L$ and belief state transformers are coalgebra of type $\mathcal{D}S \to (\mathcal{P}\mathcal{D}S)^L$. One would like to take $F = \mathcal{P}^L$ and $\mathcal{M} = \mathcal{D}$ and reuse the above construction but, unfortunately, $\mathcal{P}^L$ does *not* have a *suitable* lifting to $\mathrm{EM}(\mathcal{D})$. This is a consequence of two well known facts: the lack of a *suitable* distributive law $\rho\colon \mathcal{D}\mathcal{P} \Rightarrow \mathcal{P}\mathcal{D}$ [64] [4] and the one-to-one correspondence between distributive laws and liftings, see e.g. [31]. In the next section, we will nevertheless provide a "powerset-like" functor on $\mathrm{EM}(\mathcal{D})$ that we will exploit then in Section 6 to properly model PA as belief-state transformers.

5 Coalgebras on Convex Algebras

In this section we provide several functors on $\mathrm{EM}(\mathcal{D})$ that will be used in the modelling of probabilistic automata as coalgebras over $\mathrm{EM}(\mathcal{D})$. This will make explicit the implicit algebraic structure (convexity) in probabilistic automata and lead to distribution bisimilarity as natural semantics for probabilistic automata in Section 6.

[4] As shown in [64], there is no distributive law of the powerset monad over the distribution monad. Note that a "trivial" lifting and a corresponding distributive law of the powerset *functor* over the distribution monad exists, it is based on [11] and has been exploited in [31]. However, the corresponding "determinisation" is trivial, in the sense that its distribution bisimilarity coincides with bisimilarity, and it does not correspond to the belief-state transformer.

5.1 Convex Powerset on Convex Algebras

We now define a functor, the (nonempty) convex powerset functor, on $\mathrm{EM}(\mathcal{D})$. Let $\mathbb{A}$ be a convex algebra. We define $\mathcal{P}_c\mathbb{A}$ to be $\mathbb{A}_c = (A_c, a_c)$ where $A_c = \{C \subseteq A \mid C \neq \emptyset, C \text{ is convex}\}$ and a_c is the convex algebra structure given by the following *pointwise* binary convex combinations: $pC + \bar{p}D = \{pc + \bar{p}d \mid c \in C, d \in D\}$.

It is important that we only allow nonempty convex subsets in the carrier A_c of $\mathcal{P}_c\mathbb{A}$, as otherwise the projection axiom fails.

▶ **Lemma 14.** *$\mathcal{P}_c\mathbb{A}$ as defined above is a convex algebra.* ◀

For convex subsets of a finite dimensional vector space, the pointwise operations are known as the Minkowski addition and are a basic construction in convex geometry, see e.g. [47]. The pointwise way of defining algebras over subsets (carriers of subalgebras) has also been studied in universal algebra, see e.g. [8, 7, 9].

Next, we define $\mathcal{P}_c$ on arrows. For a convex homomorphism $h\colon \mathbb{A} \to \mathbb{B}$, $\mathcal{P}_c h = \mathcal{P}h$.

▶ **Lemma 15.** *$\mathcal{P}_c h$ is a convex algebra homomorphism from $\mathcal{P}_c\mathbb{A}$ to $\mathcal{P}_c\mathbb{B}$ if $h\colon \mathbb{A} \to \mathbb{B}$ is a convex homomorphism.* ◀

The following property is now a direct consequence of the definition of $\mathcal{P}_c$, Lemma 14, Lemma 15, and Proposition 13.

▶ **Proposition 16.** *$\mathcal{P}_c$ is a functor on $\mathrm{EM}(\mathcal{D})$.* ◀

▶ Remark 17. $\mathcal{P}_c$ is not a lifting of C to $\mathrm{EM}(\mathcal{D})$, but it holds that $C = \mathcal{U} \circ \mathcal{P}_c \circ \mathcal{F}$ as illustrated below on the left. $\mathcal{P}_c$ is also not a lifting of $\mathcal{P}_{ne}$, the nonempty powerset functor, but we have an embedding natural transformation $e\colon \mathcal{U} \circ \mathcal{P}_c \Rightarrow \mathcal{P}_{ne} \circ \mathcal{U}$ given by $e(C) = C$, i.e., we are in the situation:

$$\begin{tikzcd} \mathrm{EM}(\mathcal{D}) \arrow[r, "\mathcal{P}_c"] & \mathrm{EM}(\mathcal{D}) \arrow[d, "\mathcal{U}"] \\ \mathbf{Sets} \arrow[u, "\mathcal{F}"] \arrow[r, "C"] & \mathbf{Sets} \end{tikzcd} \qquad \begin{tikzcd} \mathrm{EM}(\mathcal{D}) \arrow[r, "\mathcal{P}_c"] \arrow[d, "\mathcal{U}"'] \arrow[dr, phantom, "\supseteq"] & \mathrm{EM}(\mathcal{D}) \arrow[d, "\mathcal{U}"] \\ \mathbf{Sets} \arrow[r, "\mathcal{P}_{ne}"] & \mathbf{Sets} \end{tikzcd}$$

The right diagram in Remark 17 simply states that every convex subset is a subset, but this fact and the natural transformation e are useful in the sequel. In particular, using e we can show the next result.

▶ **Proposition 18.** *$\mathcal{P}_c$ is a monad on $\mathrm{EM}(\mathcal{D})$, with η and μ as for the powerset monad.* ◀

5.2 Termination on Convex Algebras

The functor $\mathcal{P}_c$ defined in the previous section allows only for nonempty convex subsets. We still miss is a way to express termination.

The question of termination amounts to the question of extending a convex algebra $\mathbb{A}$ with a single element $*$. This question turns out to be rather involved, beyond the scope of this paper. The answer from [55] is: there are many ways to extend any convex algebra $\mathbb{A}$ with a single element, but there is only one natural functorial way. Somehow now mathematics is forcing us the choice of a specific computational behaviour for termination!

Given a convex algebra $\mathbb{A}$, let $\mathbb{A} + 1$ have the carrier $A + \{*\}$ for $* \notin A$ and convex operations given by

$$px \oplus \bar{p}y = \begin{cases} px + \bar{p}y\,, & x, y \in A, \\ * \qquad\quad , & x = * \text{ or } y = *. \end{cases} \tag{2}$$

Here, the newly added $*$ behaves as a black hole that attracts every other element of the algebra in a convex combination. It is worth to remark that this extension is folklore [23].

▶ **Proposition 19** ([55, 23]). *$\mathbb{A}+1$ as defined above is a convex algebra that extends $\mathbb{A}$ by a single element. The map $h+1$ obtained with the termination functor in* **Sets** *is a convex homomorphism if $h\colon \mathbb{A}\to\mathbb{B}$ is. The assignments $(-)+1$ give a functor on* $\mathrm{EM}(\mathcal{D})$. ◀

We call the functor $(-)+1$ on $\mathrm{EM}(\mathcal{D})$ the termination functor, due to the following.

▶ **Lemma 20.** *The functor $(-)+1$ is a lifting of the termination functor to* $\mathrm{EM}(\mathcal{D})$. ◀

▶ Remark 21. Note that we are abusing notation here: Our termination functor $(-)+1$ on $\mathrm{EM}(\mathcal{D})$ is not the coproduct $(-)+1$ in $\mathrm{EM}(\mathcal{D})$. The coproduct was concretely described in [32, Lemma 4], and the coproduct $\mathbb{X}+1$ has a much larger carrier than $X+1$. Nevertheless, we use the same notation as it is very intuitive and due to Lemma 20.

5.3 Constant Exponent on Convex Algebras

We now show the existence of a constant exponent functor on $\mathrm{EM}(\mathcal{D})$. Let L be a set of labels or actions. Let $\mathbb{A}$ be a convex algebra. Consider $\mathbb{A}^L$ with carrier $A^L = \{f \mid f\colon L\to A\}$ and operations defined (pointwise) by $(pf+\bar{p}g)(l) = pf(l)+\bar{p}g(l)$. The following property follows directly from the definitions.

▶ **Proposition 22.** *$\mathbb{A}^L$ is a convex algebra. If $h\colon\mathbb{A}\to\mathbb{B}$ is a convex homomorphism, then so is $h^L\colon \mathbb{A}^L\to\mathbb{B}^L$ defined as in* **Sets**. *Hence, $(-)^L$ defined above is a functor on* $\mathrm{EM}(\mathcal{D})$. ◀

We call $(-)^L$ the constant exponent functor on $\mathrm{EM}(\mathcal{D})$. The name and the notation is justified by the following (obvious) property.

▶ **Lemma 23.** *The constant exponent $(-)^L$ on $\mathrm{EM}(\mathcal{D})$ is a lifting of the constant exponent functor $(-)^L$ on* **Sets**. ◀

▶ **Example 24.** Consider a free algebra $\mathcal{F}S = (\mathcal{D}S,\mu)$ of distributions over the set S. By applying first the functor $\mathcal{P}_\mathrm{c}$, then $(-)+1$ and then $(-)^L$, one obtains the algebra

$$(\mathcal{P}_\mathrm{c}\mathcal{F}S+1)^L = \begin{pmatrix} \mathcal{D}\left((CS+1)^L\right) \\ \downarrow\alpha \\ (CS+1)^L \end{pmatrix}$$

where CS is the set of non-empty convex subsets of distributions over S, and α corresponds to the convex operations[5] $\sum p_i f_i$ defined by

$$\left(\sum p_i f_i\right)(l) = \begin{cases} \{\sum p_i\xi_i \mid \xi_i\in f_i(l)\} & f_i(l)\in CS \text{ for all } i\in\{1,\dots,n\} \\ * & f_i(l) = * \text{ for some } i\in\{1,\dots,n\} \end{cases}$$

5.4 Transition Systems on Convex Algebras

We now compose the three functors introduced above to properly model transition systems as coalgebras on $\mathrm{EM}(\mathcal{D})$. The functor that we are interested in is $(\mathcal{P}_\mathrm{c}+1)^L\colon \mathrm{EM}(\mathcal{D})\to\mathrm{EM}(\mathcal{D})$. A coalgebra $(\mathbb{S},c)$ for this functor can be thought of as a transition system with labels in L where the state space carries a convex algebra and the transition function $c\colon\mathbb{S}\to(\mathcal{P}_\mathrm{c}\mathbb{S}+1)^L$ is

[5] In this case, for future reference, it is convenient to spell out the n-ary convex operations.

a homomorphism of convex algebras. This property entails compositionality: the transitions of a composite state $p x_1 + \bar{p} x_2$ are fully determined by the transitions of its components x_1 and x_2, as shown in the next proposition. We write $x \overset{a}{\mapsto} y$ for $x, y \in S$, the carrier of $\mathbb{S}$ if $y \in c(x)(a)$, and $x \overset{a}{\not\mapsto}$ if $c(x)(a) = *$.

▶ **Proposition 25.** *Let $(\mathbb{S}, c)$ be a $(\mathcal{P}_{\mathrm{c}}+1)^L$-coalgebra, and let $x_1, x_2, y_1, y_2, z \in S$ be elements of S, the carrier of $\mathbb{S}$. Then, for all $p \in (0,1)$, and $a \in L$*

- *$p x_1 + \bar{p} x_2 \overset{a}{\mapsto} z$ iff $z = p y_1 + \bar{p} y_2$, $x_1 \overset{a}{\mapsto} y_1$ and $x_2 \overset{a}{\mapsto} y_2$;*
- *$p x_1 + \bar{p} x_2 \overset{a}{\not\mapsto}$ iff $x_1 \overset{a}{\not\mapsto} y_1$ or $x_2 \overset{a}{\not\mapsto} y_2$.* ◀

Transition systems on convex algebras are the bridge between PA and LTSs. In the next section we will show that one can transform an arbitrary PA into a $(\mathcal{P}_{\mathrm{c}}+1)^L$-coalgebra and that, in the latter, behavioural equivalence coincides with the standard notion of bisimilarity for LTSs (Corollary 30).

6 From PA to Belief-State Transformers

Before turning our attention to PA, it is worth to make a further step of abstraction.

Recall from Remark 17 how $\mathcal{P}_{\mathrm{c}}$ is related to C and $\mathcal{P}_{ne}$. The following definition is the obvious generalisation.

▶ **Definition 26.** Let $\mathcal{M}\colon \mathbf{Sets} \to \mathbf{Sets}$ be a monad and $\mathcal{L}_1, \mathcal{L}_2\colon \mathbf{Sets} \to \mathbf{Sets}$ be two functors. A functor $\mathcal{H}\colon \mathrm{EM}(\mathcal{M}) \to \mathrm{EM}(\mathcal{M})$ is

- a *quasi lifting* of $\mathcal{L}_1$ if the diagram on the left commutes.
- a *lax lifting* of $\mathcal{L}_2$ if there exists an injective natural transformation $e\colon \mathcal{U} \circ \mathcal{H} \Rightarrow \mathcal{L}_2 \circ \mathcal{U}$ as depicted on the right.
- an $(\mathcal{L}_1, \mathcal{L}_2)$ *quasi-lax lifting* if it is both a quasi lifting of $\mathcal{L}_1$ and a lax lifting of $\mathcal{L}_2$.

$$\begin{tikzcd} \mathrm{EM}(\mathcal{M}) \arrow[r, "\mathcal{H}"] & \mathrm{EM}(\mathcal{M}) \arrow[d, "\mathcal{U}"] \\ \mathbf{Sets} \arrow[u, "\mathcal{F}"] \arrow[r, "\mathcal{L}_1"] & \mathbf{Sets} \end{tikzcd} \qquad \begin{tikzcd} \mathrm{EM}(\mathcal{M}) \arrow[r, "\mathcal{H}"] \arrow[d, "\mathcal{U}"'] \arrow[dr, phantom, "\Swarrow"] & \mathrm{EM}(\mathcal{M}) \arrow[d, "\mathcal{U}"] \\ \mathbf{Sets} \arrow[r, "\mathcal{L}_2"] & \mathbf{Sets} \end{tikzcd}$$ ◇

So, for instance, $\mathcal{P}_{\mathrm{c}}$ is a $(C, \mathcal{P}_{ne})$ quasi-lax lifting. From this fact, it follows that $(\mathcal{P}_{\mathrm{c}}+1)^L$ is a $((C+1)^L, (\mathcal{P}_{ne}+1)^L)$ quasi-lax lifting. Another interesting example is the generalised determinisation (Section 4.2): it is easy to see that $\overline{F}$ is a $(F\mathcal{M}, F)$-quasi-lax lifting. Indeed, like in the generalised powerset construction, one can construct the following functors.

$$\begin{tikzcd} & \mathrm{Coalg}_{\mathrm{EM}(\mathcal{M})}(\mathcal{H}) \arrow[dr, "\overline{\mathcal{U}}"] & \\ \mathrm{Coalg}_{\mathbf{Sets}}(\mathcal{L}_1) \arrow[ur, "\overline{\mathcal{F}}"] & & \mathrm{Coalg}_{\mathbf{Sets}}(\mathcal{L}_2) \end{tikzcd}$$

We first define $\overline{\mathcal{F}}$. Take an $\mathcal{L}_1$-coalgebra (S, c) and recall that $\mathcal{F}S$ is the free algebra $\mu\colon \mathcal{M}\mathcal{M}S \to \mathcal{M}S$. The left diagram in Definition 26 entails that $\mathcal{H}\mathcal{F}S$ is an algebra $\alpha\colon \mathcal{M}\mathcal{L}_1 S \to \mathcal{L}_1 S$. We call $\mathcal{U}c^\sharp$ the composition $\mathcal{U}\mathcal{F}S = \mathcal{M}S \xrightarrow{\mathcal{M}c} \mathcal{M}\mathcal{L}_1 S \xrightarrow{\alpha} \mathcal{L}_1 S = \mathcal{U}\mathcal{H}\mathcal{F}S$. The next lemma shows that $c^\sharp\colon \mathcal{F}S \to \mathcal{H}\mathcal{F}S$ is a map in $\mathrm{EM}(\mathcal{M})$.

▶ **Lemma 27.** *There is a 1-1 correspondence between $\mathcal{L}_1$-coalgebras on* **Sets** *and $\mathcal{H}$-coalgebras on $\mathrm{EM}(\mathcal{M})$ with carriers free algebras:*

$$\frac{c\colon S \to \mathcal{L}_1 S \quad \textit{in} \quad \mathbf{Sets}}{\overline{\overline{c^\#\colon \mathcal{F}S \to \mathcal{H}\mathcal{F}S \quad \textit{in} \quad \mathrm{EM}(\mathcal{M})}}}$$

$M = (S, L, \to)$ $(S, c_M : S \to (\mathcal{P}\mathcal{D})^L)$ (S, c_M) c_M	$M_c = (S, L, \to_c)$ $(S, \bar{c}_M : S \to (C+1)^L)$ $(S, \bar{c}_M) = \mathcal{T}_{\mathrm{conv}}(S, c_M)$ $\bar{c}_M = \mathrm{conv}^L \circ c_M$	$M_{bs} = (\mathcal{D}S, L, \mapsto)$ $(\mathcal{D}S, \hat{c}_M : \mathcal{D}S \to (\mathcal{P}\mathcal{D}S)^L)$ $(S, \hat{c}_M) = \overline{\mathcal{U}} \circ \overline{\mathcal{F}} \circ \mathcal{T}_{\mathrm{conv}}(S, c_M)$ $\hat{c}_M = (e_{\mathcal{F}S} + 1)^L \circ \mathcal{U}\bar{c}_M^\sharp$

Table 1 The three PA models, their corresponding **Sets**-coalgebras, and relations to M.

- *given c, we have $\mathcal{U}c^{\#} = \alpha \circ \mathcal{M}c$ for $\alpha = \mathcal{H}\mathcal{F}S$,*
- *given $c^{\#}$, we have $c = \mathcal{U}c^{\#} \circ \eta$.*

The assignment $\overline{\mathcal{F}}(S, c) = (\mathcal{F}S, c^{\#})$ and $\overline{\mathcal{F}}(h) = \mathcal{M}h$ gives a functor $\overline{\mathcal{F}}\colon \mathrm{Coalg}_{\mathbf{Sets}}(\mathcal{L}_1) \to \mathrm{Coalg}_{\mathrm{EM}(\mathcal{M})}(\mathcal{H})$. ◀

Now we can define $\overline{\mathcal{U}}\colon \mathrm{Coalg}_{\mathrm{EM}(\mathcal{M})}(\mathcal{H}) \to \mathrm{Coalg}_{\mathbf{Sets}}(\mathcal{L}_2)$ as mapping every coalgebra $(\mathbb{S}, c)$ with $c\colon \mathbb{S} \to \mathcal{H}\mathbb{S}$ into

$$\overline{\mathcal{U}}(\mathbb{S}, c) = (\mathcal{U}\mathbb{S}, e_{\mathbb{S}} \circ \mathcal{U}c) \quad \text{where} \quad \mathcal{U}\mathbb{S} \xrightarrow{\mathcal{U}c} \mathcal{U}\mathcal{H}\mathbb{S} \xrightarrow{e_{\mathbb{S}}} \mathcal{L}_2\mathcal{U}\mathbb{S}$$

and every coalgebra homomorphism $h\colon (\mathbb{S}, c) \to (\mathbb{T}, d)$ into $\overline{\mathcal{U}}h = \mathcal{U}h$. Routine computations confirm that $\overline{\mathcal{U}}$ is a functor.

Since $\overline{\mathcal{U}}$ is a functor every kernel bisimulation on $(\mathbb{S}, c)$ is also a kernel bisimulation on $\overline{\mathcal{U}}(\mathbb{S}, c)$. The converse is not true in general: a kernel bisimulation R on $\overline{\mathcal{U}}(\mathbb{S}, c)$ is a kernel bisimulation on $(\mathbb{S}, c)$ only if it is a *congruence* with respect to the algebraic structure of $\mathbb{S}$. Formally, R is a congruence if and only if the set $\mathcal{U}\mathbb{S}/R$ of equivalence classes of R carries an Eilenberg-Moore algebra and the function $\mathcal{U}[-]_R\colon \mathcal{U}\mathbb{S} \to \mathcal{U}\mathbb{S}/R$ mapping every element of $\mathcal{U}\mathbb{S}$ to its R-equivalence class is an algebra homomorphism.

▶ **Proposition 28.** *The following are equivalent:*
- *R is a kernel bisimulation on $(\mathbb{S}, c)$,*
- *R is a congruence of $\mathbb{S}$ and a kernel bisimulation of $\overline{\mathcal{U}}(\mathbb{S}, c)$.* ◀

In particular, Proposition 28 and the following result ensure that the functor $\overline{\mathcal{U}}\colon \mathrm{Coalg}_{\mathrm{EM}(\mathcal{D})}(\mathcal{P}_c + 1)^L \to \mathrm{Coalg}_{\mathbf{Sets}}\,\mathcal{P}^L$ preserves and reflect $\approx$.

▶ **Proposition 29.** *Let $(\mathbb{S}, c)$ be a $(\mathcal{P}_c + 1)^L$-coalgebra. Behavioural equivalence on $\overline{\mathcal{U}}(\mathbb{S}, c)$ is a convex congruence*[6]*.* ◀

▶ **Corollary 30.** *The functor $\overline{\mathcal{U}}$ preserves and reflects behavioural equivalence.* ◀

This means that $\approx$ for $(\mathcal{P}_c + 1)^L$-coalgebras, called transition systems on convex algebras in Section 5.4, coincides with the standard notion of bisimilarity for LTSs.

Table 1 summarises all models of PA: from the classical model M being a $\mathcal{P}\mathcal{D}^L$-coalgebra (S, c_M) on **Sets**, via the convex model M_c obtained as $\mathcal{T}_{\mathrm{conv}}(S, c_M)$, to the belief state transformer M_{bs}. The latter coincides with $\overline{\mathcal{U}} \circ \overline{\mathcal{F}} \circ \mathcal{T}_{\mathrm{conv}}(S, c_M)$.

▶ **Theorem 31.** *Let (S, c_M) be a probabilistic automaton. For all $\xi, \zeta \in \mathcal{D}S$,*

$$\xi \sim_d \zeta \quad \Leftrightarrow \quad \xi \approx \zeta \ \text{ in } \overline{\mathcal{U}} \circ \overline{\mathcal{F}} \circ \mathcal{T}_{\mathrm{conv}}(S, c_M).$$ ◀

Hence, distribution bisimilarity is indeed behavioural equivalence on the belief-state transformer and it coincides with standard bisimilarity.

[6] Convex congruences are congruences of convex algebras, see e.g. [54]. They are convex equivalences, i.e., closed under componentwise-defined convex combinations.

7 Bisimulations Up-To Convex Hull

As we mentioned in Section 4.2, the generalised determinisation allows for the use of up-to techniques [42, 45]. An important example is shown in [6]: given a non-deterministic automaton $c\colon S \to 2 \times \mathcal{P}(S)^A$, one can reason on its determinisation $\mathcal{U}c^\sharp\colon \mathcal{P}(S) \to 2 \times \mathcal{P}(S)^A$ up-to the algebraic structure carried by the state space $\mathcal{P}(S)$. Given a probabilistic automaton $(S, L, \to)$, we would like to exploit the algebraic structure carried by $\mathcal{D}(S)$ to prove properties of the corresponding belief states transformer $(\mathcal{D}(S), L, \mapsto)$. Unfortunately, the lack of a suitable distributive law [64] makes it impossible to reuse the abstract results in [5] and all the proofs need to be done from scratch by adapting the theory in [45] to the case of probabilistic automata.

Hereafter we fix a PA $M = (S, L, \to)$ and the corresponding belief states transformer $M_{bs} = (\mathcal{D}(S), L, \mapsto)$. We denote by $Rel_{\mathcal{D}(S)}$ the lattice of relations over $\mathcal{D}(S)$ and define the monotone function $\mathrm{b}\colon Rel_{\mathcal{D}(S)} \to Rel_{\mathcal{D}(S)}$ mapping every relation $R \in Rel_{\mathcal{D}(S)}$ into

$$\begin{aligned}\mathrm{b}(R) \quad ::= \quad \{(\zeta_1, \zeta_2) \quad | \quad \forall a \in L, \quad & \forall \zeta_1' \text{ s.t. } \zeta_1 \overset{a}{\mapsto} \zeta_1',\ \exists \zeta_2' \text{ s.t. } \zeta_2 \overset{a}{\mapsto} \zeta_2' \text{ and } (\zeta_1', \zeta_2') \in R,\\ & \forall \zeta_1' \text{ s.t. } \zeta_2 \overset{a}{\mapsto} \zeta_2',\ \exists \zeta_1' \text{ s.t. } \zeta_1 \overset{a}{\mapsto} \zeta_1' \text{ and } (\zeta_1', \zeta_2') \in R\}.\end{aligned}$$

A *bisimulation* is a relation R such that $R \subseteq \mathrm{b}(R)$. Observe that these are just regular bisimulations for labeled transition systems and that the greatest fixpoint of b coincides exactly with $\sim_d$. The *coinduction* principle informs us that to prove that $\zeta_1 \sim_d \zeta_2$ it is enough to exhibit a bisimulation R such that $(\zeta_1, \zeta_2) \in R$.

▶ **Example 32.** Consider the PA in Figure 1 (left) and the belief-state transformer generated by it (right). It is easy to see that the (Dirac distributions of the) states x_2 and y_2 are in $\sim_d$: the relation $\{(x_2, y_2)\}$ is a bisimulation. Also $\{(x_3, y_3)\}$ is a bisimulation: both $x_3 \not\overset{a}{\mapsto}$ and $y_3 \not\overset{a}{\mapsto}$. More generally, for all $\zeta, \xi \in \mathcal{D}(S)$, $p, q \in [0, 1]$, $p\zeta + \bar{p}x_3 \sim_d q\xi + \bar{q}y_3$ since both

$$p\zeta + \bar{p}x_3 \not\overset{a}{\mapsto} \text{ and } q\xi + \bar{q}y_3 \not\overset{a}{\mapsto}\ . \tag{3}$$

Proving that $x_0 \sim_d y_0$ is more complicated. We will show this in Example 35 but, for the time being, observe that one would need an infinite bisimulation containing the following pairs of states.

$$\begin{tikzcd}[column sep=small]
x_0 \arrow[r, mapsto, "a"] \arrow[d, dotted, no head] & x_1 \arrow[r, mapsto, "a"] \arrow[d, dotted, no head] & \frac{1}{2}x_1 + \frac{1}{2}x_2 \arrow[r, mapsto, "a"] \arrow[d, dotted, no head] & \frac{1}{4}x_1 + \frac{3}{4}x_2 \arrow[r, mapsto, "a"] \arrow[d, dotted, no head] & \dots \arrow[d, dotted, no head] \\
y_0 \arrow[r, mapsto, "a"] & \frac{1}{2}y_1 + \frac{1}{2}y_2 \arrow[r, mapsto, "a"] & \frac{1}{4}y_1 + \frac{3}{4}y_2 \arrow[r, mapsto, "a"] & \frac{1}{8}y_1 + \frac{7}{8}y_2 \arrow[r, mapsto, "a"] & \dots
\end{tikzcd}$$

Indeed, all the distributions depicted above have infinitely many possible choices for $\overset{a}{\mapsto}$ but, whenever one of them executes a depicted transition, the corresponding distribution is forced, because of (3), to also choose the depicted transition.

An *up-to technique* is a monotone map $\mathrm{f}\colon Rel_{\mathcal{D}(S)} \to Rel_{\mathcal{D}(S)}$, while a *bisimulation up-to f* is a relation R such that $R \subseteq \mathrm{b}\,\mathrm{f}(R)$. An up-to technique f is said to be *sound* if, for all $R \in Rel_{\mathcal{D}(S)}$, $R \subseteq \mathrm{b}\,\mathrm{f}(R)$ entails that $R \subseteq \sim_d$. It is said to be *compatible* if $\mathrm{f}\,\mathrm{b}(R) \subseteq \mathrm{b}\,\mathrm{f}(R)$. In [45], it is shown that every compatible up-to technique is also sound.

Hereafter we consider the *convex hull* technique $\mathrm{conv}\colon Rel_{\mathcal{D}(S)} \to Rel_{\mathcal{D}(S)}$ mapping every relation $R \in Rel_{\mathcal{D}(S)}$ into its convex hull which, for the sake of clarity, is

$$\mathrm{conv}(R) = \{(p\zeta_1 + \bar{p}\xi_1, p\zeta_2 + \bar{p}\xi_2) \mid (\zeta_1, \zeta_2) \in R,\ (\xi_1, \xi_2) \in R \text{ and } p \in [0, 1]\}.$$

▶ **Proposition 33.** conv *is compatible.* ◀

This result has two consequences: first, conv is sound and thus one can prove $\sim_d$ by means of bisimulation up-to conv; second, conv can be effectively combined with other compatible up-to techniques (for more details see [45] or Appendix C). In particular, by combining conv with up-to equivalence – which is well known to be compatible – one obtains up-to congruence $\mathrm{cgr}\colon Rel_{\mathcal{D}(S)} \to Rel_{\mathcal{D}(S)}$. This technique maps a relation R into its congruence closure: the smallest relation containing R which is a congruence.

▶ **Proposition 34.** cgr *is compatible.* ◀

Since cgr is compatible and thus sound, we can use bisimulation up-to cgr to check $\sim_d$.

▶ **Example 35.** We can now prove that, in the PA depicted in Figure 1, $x_0 \sim_d y_0$. It is easy to see that the relation $R = \{(x_2, y_2), (x_3, y_3), (x_1, \frac{1}{2}y_1 + \frac{1}{2}y_2), (x_0, y_0)\}$ is a bisimulation up-to cgr: consider $(x_1, \frac{1}{2}y_1 + \frac{1}{2}y_2)$ (the other pairs are trivial) and observe that

$$\begin{tikzcd} x_1 \arrow[r, mapsto, "a"] \arrow[d, dash, dotted, "R"'] & \frac{1}{2}x_1 + \frac{1}{2}x_2 \arrow[d, dash, dotted, "\mathrm{cgr}(R)"] \\ \frac{1}{2}y_1 + \frac{1}{2}y_2 \arrow[r, mapsto, "a"'] & \frac{1}{4}y_1 + \frac{3}{4}y_2 \end{tikzcd} \qquad \begin{tikzcd} x_1 \arrow[r, mapsto, "a"] \arrow[d, dash, dotted, "R"'] & \frac{1}{2}x_3 + \frac{1}{2}x_2 \arrow[d, dash, dotted, "\mathrm{cgr}(R)"] \\ \frac{1}{2}y_1 + \frac{1}{2}y_2 \arrow[r, mapsto, "a"'] & \frac{1}{2}y_3 + \frac{1}{2}y_2 \end{tikzcd}$$

Since all the transitions of x_1 and $\frac{1}{2}y_1 + \frac{1}{2}y_2$ are obtained as convex combination of the two above, the arriving states are forced to be in $\mathrm{cgr}(R)$. In symbols, if $x_1 \overset{a}{\mapsto} \zeta = p(\frac{1}{2}x_1 + \frac{1}{2}x_2) + \bar{p}(\frac{1}{2}x_3 + \frac{1}{2}x_2)$, then $\frac{1}{2}y_1 + \frac{1}{2}y_2 \overset{a}{\mapsto} \xi = p(\frac{1}{4}y_1 + \frac{3}{4}y_2) + \bar{p}(\frac{1}{2}y_3 + \frac{1}{2}y_2)$ and $(\zeta, \xi) \in \mathrm{cgr}(R)$.

Recall that in Example 32, we showed that to prove $x_0 \sim_d y_0$ without up-to techniques one would need an infinite bisimulation. Instead, the relation R in Example 35 is a *finite* bisimulation up-to cgr. It turns out that this holds in general: one can check $\sim_d$ by means of only finite bisimulations up-to. The key to this result is the following theorem, recently proved in [54].

▶ **Theorem 36.** *Every congruence of a finitely generated convex algebra is finitely generated.*

This result informs us that for a PA with a finite state space S, $\sim_d \subseteq \mathcal{D}(S) \times \mathcal{D}(S)$ is finitely generated (since $\sim_d$ is a congruence, see Proposition 29). In other words there exists a finite relation R such that $\mathrm{cgr}(R) = \sim_d$. Such R is a finite bisimulation up-to cgr:

$$R \subseteq \mathrm{cgr}(R) = \sim_d = \mathrm{b}(\sim_d) = \mathrm{b}(\mathrm{cgr}(R)).$$

▶ **Corollary 37.** *Let $(S, L, \to)$ be a finite PA and let $\zeta_1, \zeta_2 \in \mathcal{D}(S)$ be two distributions such that $\zeta_1 \sim_d \zeta_2$. Then, there exists a finite bisimulation up-to* cgr *R such that $(\zeta_1, \zeta_2) \in R$.* ◀

8 Conclusions and Future Work

Belief-state transformers and distribution bisimilarity have a strong coalgebraic foundation which leads to a new proof method – bisimulation up-to convex hull. More interestingly, and somewhat surprisingly, proving distribution bisimilarity can be achieved using only *finite* bisimulation up-to witness. This opens exciting new avenues: Corollary 37 gives us hope that bisimulations up-to may play an important role in designing algorithms for automatic equivalence checking of PA, similar to the one played for NFA. Exploring their connections with the algorithms in [26, 20] is our next step.

From a more abstract perspective, our work highlights some limitations of the bialgebraic approach [61, 3, 34]. Despite the fact that our structures are coalgebras on algebras, they are not bialgebras: but still $\approx$ is a congruence and it is amenable to up-to techniques. We believe that *lax* bialgebra may provide some deeper insights.

References

1. Manindra Agrawal, S. Akshay, Blaise Genest, and P. S. Thiagarajan. Approximate verification of the symbolic dynamics of markov chains. In *Proc LICS 2012*, pages 55–64. IEEE, 2012.
2. Christel Baier and Joost-Pieter Katoen. *Principles of model checking.* MIT Press, 2008.
3. F. Bartels. *On generalised coinduction and probabilistic specification formats: distributive laws in coalgebraic modelling.* PhD thesis, Vrije Universiteit, Amsterdam, 2004.
4. F. Bartels, A. Sokolova, and E.P. de Vink. A hierarchy of probabilistic system types. *Theoretical Computer Science*, 327:3–22, 2004.
5. Filippo Bonchi, Daniela Petrişan, Damien Pous, and Jurriaan Rot. Coinduction up-to in a fibrational setting. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, page 20. ACM, 2014.
6. Filippo Bonchi and Damien Pous. Checking nfa equivalence with bisimulations up to congruence. In *Principle of Programming Languages (POPL)*, pages 457–468. ACM, 2013.
7. Ivica Bošnjak and Rozália Madarász. On power structures. *Algebra and Discrete Mathematics*, 2:14–35, 2003.
8. Ivica Bošnjak and Rozália Madarász. Some results on complex algebras of subalgebras. *Novi Sad Journal of Mathematics*, 237(2):231–240, 2007.
9. C. Brink. On power structures. *Algebra Universalis*, 30:177–216, 1993.
10. P. Castro, P. Panangaden, and D. Precup. Equivalence relations in fully and partially observable Markov decision processes. In *Proc. IJCAI 2009*, pages 1653–1658, 2009.
11. S. Crafa and F. Ranzato. A spectrum of behavioral relations over LTSs on probability distributions. In *Proc. CONCUR 2011*, pages 124–139. LNCS 6901, 2011.
12. Fredrik Dahlqvist, Vincent Danos, and Ilias Garnier. Robustly parameterised higher-order probabilistic models. In Josée Desharnais and Radha Jagadeesan, editors, *27th International Conference on Concurrency Theory, CONCUR 2016, August 23-26, 2016, Québec City, Canada*, volume 59 of *LIPIcs*, pages 23:1–23:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.
13. Y. Deng, R. van Glabbeek, M. Hennessy, and C. Morgan. Characterising testing preorders for finite probabilistic processes. *Logical Methods in Computer Science*, 4(4), 2008.
14. Yuxin Deng and Matthew Hennessy. On the semantics of Markov automata. *Inf. Comput.*, 222:139–168, 2013.
15. Yuxin Deng, Rob J. van Glabbeek, Matthew Hennessy, and Carroll Morgan. Testing finitary probabilistic processes. In *Proc. CONCUR 2009*, volume 5710 of *LNCS*, pages 274–288, 2009.
16. Ernst-Erich Doberkat. Eilenberg-Moore algebras for stochastic relations. *Inform. and Comput.*, 204(12):1756–1781, 2006.
17. Ernst-Erich Doberkat. Erratum and addendum: Eilenberg-Moore algebras for stochastic relations [mr2277336]. *Inform. and Comput.*, 206(12):1476–1484, 2008.
18. Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. Equivalence of labeled Markov chains. *Int. J. Found. Comput. Sci.*, 19(3):549–563, 2008.
19. Laurent Doyen, Thierry Massart, and Mahsa Shirmohammadi. Limit synchronization in markov decision processes. In *Proc FOSSACS 2014*, volume 8412 of *LNCS*, pages 58–72, 2014.
20. Christian Eisentraut, Holger Hermanns, Julia Krämer, Andrea Turrini, and Lijun Zhang. Deciding bisimilarities on distributions. In *QEST'13*, volume 8054 of *LNCS*, pages 72–88. Springer, 2013.
21. Christian Eisentraut, Holger Hermanns, and Lijun Zhang. On probabilistic automata in continuous time. In *LICS'10*, pages 342–351. IEEE Computer Society, 2010.

22 Yuan Feng and Lijun Zhang. When equivalence and bisimulation join forces in probabilistic automata. In *Proc. FM 2014*, volume 8442 of *LNCS*, pages 247–262, 2014.

23 Tobias Fritz. Convex spaces I: Definition and Examples. arXiv:0903.5522v3 [math.MG].

24 Noah D. Goodman, Vikash K. Mansinghka, Daniel M. Roy, Keith Bonawitz, and Joshua B. Tenenbaum. Church: a language for generative models. *CoRR*, abs/1206.3255, 2012.

25 Matthew Hennessy. Exploring probabilistic bisimulations, part I. *Formal Asp. Comput.*, 24(4-6):749–768, 2012.

26 Holger Hermanns, Jan Krcál, and Jan Kretínský. Probabilistic bisimulation: Naturally on distributions. In *Proc. CONCUR'14*, volume 8704 of *LNCS*, pages 249–265, 2014.

27 Chris Heunen, Ohad Kammar, Sam Staton, and Hongseok Yang. A convenient category for higher-order probability theory. *CoRR*, abs/1701.02547, 2017.

28 B. Jacobs. Convexity, duality and effects. In *Theoretical computer science*, volume 323 of *IFIP Adv. Inf. Commun. Technol.*, pages 1–19. Springer, Berlin, 2010.

29 Bart Jacobs. Coalgebraic trace semantics for combined possibilitistic and probabilistic systems. *Electr. Notes Theor. Comput. Sci.*, 203(5):131–152, 2008.

30 Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, volume 59 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2016.

31 Bart Jacobs, Alexandra Silva, and Ana Sokolova. Trace semantics via determinization. *J. Comput. Syst. Sci.*, 81(5):859–879, 2015.

32 Bart Jacobs, Bas Westerbaan, and Bram Westerbaan. States of convex sets. In *Proc. FOSSACS*, volume 9034 of *LNCS*, pages 87–101, 2015.

33 B.P.F. Jacobs and J.J.M.M. Rutten. A tutorial on (co)algebras and (co)induction. *Bulletin of the EATCS*, 62:222–259, 1996.

34 Bartek Klin. Bialgebras for structural operational semantics: An introduction. *Theor. Comput. Sci.*, 412(38):5043–5069, 2011.

35 Vijay Anand Korthikanti, Mahesh Viswanathan, Gul Agha, and YoungMin Kwon. Reasoning about mdps as transformers of probability distributions. In *Proc. QEST 2010*, pages 199–208. IEEE, 2010.

36 A. Kurz. *Logics for Coalgebras and Applications to Computer Science*. PhD thesis, Ludwig-Maximilians-Universität München, 2000.

37 Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of probabilistic real-time systems. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer, 2011.

38 Marta Z. Kwiatkowska, Gethin Norman, Roberto Segala, and Jeremy Sproston. Automatic verification of real-time systems with discrete probability distributions. *Theor. Comput. Sci.*, 282(1):101–150, 2002.

39 K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94:1–28, 1991.

40 Axel Legay, Andrzej S. Murawski, Joël Ouaknine, and James Worrell. On automated verification of probabilistic programs. In C. R. Ramakrishnan and Jakob Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, volume 4963 of *Lecture Notes in Computer Science*, pages 173–187. Springer, 2008.

41 Hua Mao, Yingke Chen, Manfred Jaeger, Thomas D. Nielsen, Kim G. Larsen, and Brian Nielsen. Learning deterministic probabilistic automata from a model checking perspective. *Machine Learning*, 105(2):255–299, 2016.

42 Robin Milner. *Communication and concurrency*, volume 84. Prentice hall New York etc., 1989.

43 Matteo Mio. Upper-expectation bisimilarity and łukasiewicz μ-calculus. In *Proc. FOSSACS'14*, volume 8412 of *LNCS*, pages 335–350, 2014.

44 Michael W. Mislove. Discrete random variables over domains, revisited. In Thomas Gibson-Robinson, Philippa J. Hopcroft, and Ranko Lazic, editors, *Concurrency, Security, and Puzzles - Essays Dedicated to Andrew William Roscoe on the Occasion of His 60th Birthday*, volume 10160 of *Lecture Notes in Computer Science*, pages 185–202. Springer, 2017.

45 Damien Pous and Davide Sangiorgi. Enhancements of the coinductive proof method. In Davide Sangiorgi and Jan Rutten, editors, *Advanced Topics in Bisimulation and Coinduction*. Cambridge University Press, 2011.

46 J.J.M.M. Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249:3–80, 2000.

47 R. Schneider. *Convex bodies: the Brunn-Minkowski theory*, volume 44 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1993.

48 R. Segala. *Modeling and verification of randomized distributed real-time systems*. PhD thesis, MIT, 1995.

49 R. Segala and N.A. Lynch. Probabilistic simulations for probabilistic processes. In *Proc. CONCUR'94*, pages 481–496. LNCS 836, 1994.

50 Zbigniew Semadeni. *Monads and their Eilenberg-Moore algebras in functional analysis*. Queen's University, Kingston, Ont., 1973. Queen's Papers in Pure and Applied Mathematics, No. 33.

51 Falak Sher and Joost-Pieter Katoen. Compositional abstraction techniques for probabilistic automata. In Jos C. M. Baeten, Thomas Ball, and Frank S. de Boer, editors, *Theoretical Computer Science - 7th IFIP TC 1/WG 2.2 International Conference, TCS 2012, Amsterdam, The Netherlands, September 26-28, 2012. Proceedings*, volume 7604 of *Lecture Notes in Computer Science*, pages 325–341. Springer, 2012.

52 A. Silva, F. Bonchi, M. Bonsangue, and J. Rutten. Generalizing the powerset construction, coalgebraically. In *Proc. FSTTCS 2010*, volume 8 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 272–283, 2010.

53 Ana Sokolova. Probabilistic systems coalgebraically: A survey. *Theor. Comput. Sci.*, 412(38):5095–5110, 2011.

54 Ana Sokolova and Harald Woracek. Congruences of convex algebras. *Journal of Pure and Applied Algebra*, 219(8):3110–3148, 2015.

55 Ana Sokolova and Harald Woracek. Termination in convex sets of distributions, 2017. submitted.

56 Sam Staton. Relating coalgebraic notions of bisimulation. *Logical Methods in Computer Science*, 7(1), 2011.

57 Sam Staton. Commutative semantics for probabilistic programming. In Hongseok Yang, editor, *Programming Languages and Systems - 26th European Symposium on Programming, ESOP 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, volume 10201 of *Lecture Notes in Computer Science*, pages 855–879, 2017.

58 Sam Staton, Hongseok Yang, Frank Wood, Chris Heunen, and Ohad Kammar. Semantics for probabilistic programming: higher-order functions, continuous distributions, and soft constraints. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 525–534. ACM, 2016.

59 M.H. Stone. Postulates for the barycentric calculus. *Ann. Mat. Pura Appl. (4)*, 29:25–30, 1949.

60 T. Świrszcz. Monadic functors and convexity. *Bull. Acad. Polon. Sci. Sér. Sci. Math. Astronom. Phys.*, 22:39–42, 1974.

61 Daniele Turi and Gordon D. Plotkin. Towards a mathematical operational semantics. In *LICS 1997*, pages 280–291. IEEE Computer Society, 1997.

62 D. Varacca. *Probability, Nondeterminism and Concurrency: Two Denotational Models for Probabilistic Computation.* PhD thesis, Univ. Aarhus, 2003. BRICS Dissertation Series, DS-03-14.

63 D. Varacca and G. Winskel. Distributing probabililty over nondeterminism. *MSCS*, 16(1):87–113, 2006.

64 D. Varacca and G. Winskel. Distributing probability over non-determinism. *Mathematical Structures in Computer Science*, 16(1):87–113, 2006.

65 E.P. de Vink and J.J.M.M. Rutten. Bisimulation for probabilistic transition systems: a coalgebraic approach. *Theoretical Computer Science*, 221:271–293, 1999.

66 Ralf Wimmer, Nils Jansen, Andreas Vorpahl, Erika Ábrahám, Joost-Pieter Katoen, and Bernd Becker. High-level counterexamples for probabilistic automata. *Logical Methods in Computer Science*, 11(1), 2015.

67 U. Wolter. On corelations, cokernels, and coequations. *Electronic Notes in Theoretical Computer Science*, 33, 2000.

A Proofs for Section 5

Proof of Lemma 14. Due to Proposition 7, all we need is to check (1) idempotence, (2) parametric commutativity, and (3) parametric associativity.

(1) $C \subseteq pC + \bar{p}C$ as $c = pc + \bar{p}c \in pC + \bar{p}C$. For the oposite inclusion, consider $pc_1 + \bar{p}c_2 \in pC + \bar{p}C$ for $c_1, c_2 \in C$. As C is convex, $pc_1 + \bar{p}c_2 \in C$. Hence idempotence holds.

(2) Follows from parametric commutativity in $\mathbb{A}$. We have

$$pC + \bar{p}D = \{pc + \bar{p}d \mid c \in C, d \in D\} = \{\bar{p}d + pc \mid d \in D, c \in C\} = \bar{p}D + pC$$

proving parametric commutativity.

(3) Similarly, parametric associativity follows from parametric associativity in $\mathbb{A}$:

$$\begin{aligned} p(qC + \bar{q}D) + \bar{p}E &= \{p(qc + \bar{q}d) + \bar{p}e \mid c \in C, d \in D, e \in E\} \\ &= \{pqc + \overline{pq}\left(\frac{p\bar{q}}{\overline{pq}}d + \frac{\bar{p}}{\overline{pq}}e\right) \mid c \in C, d \in D, e \in E\} \\ &= pqC + \overline{pq}\left(\frac{p\bar{q}}{\overline{pq}}D + \frac{\bar{p}}{\overline{pq}}E\right). \end{aligned}$$

◀

Proof of Lemma 15. We have, straightforwardly,

$$\begin{aligned} \mathcal{P}_c h(pC + \bar{p}D) &= h(pC + \bar{p}D) \\ &= h(\{pc + \bar{p}d \mid c \in C, d \in D\}) \\ &= \{h(pc + \bar{p}d) \mid c \in C, d \in D\} \\ &= \{ph(c) + \bar{p}h(d) \mid c \in C, d \in D\} \\ &= ph(C) + \bar{p}h(D) \\ &= p\mathcal{P}_c h(C) + \bar{p}\mathcal{P}_c h(D). \end{aligned}$$

◀

Proof of Proposition 18. Let $\mathbb{X}$ be a convex algebra and consider $\mathcal{P}_c\mathbb{X}$. We have $\eta(x) = \{x\}$ is a convex subset, as every singleton is. Moreover, η is a convex homomorphism as $p\{x\} + \bar{p}\{y\} = \{px + \bar{p}y\}$. We have η (of $\mathcal{P}_c$) is natural if and only if the upper square of the left diagram below commutes.

$$\begin{tikzcd}
\mathcal{U}\mathbb{X} \arrow[r, "\mathcal{U}f"] \arrow[d, "\eta_{\mathbb{X}}"] \arrow[dd, bend right, "\eta_X"'] & \mathcal{U}\mathbb{Y} \arrow[d, "\eta_{\mathbb{Y}}"'] \arrow[dd, bend left, "\eta_Y"] \\
\mathcal{U}\mathcal{P}_c\mathbb{X} \arrow[r, "\mathcal{U}\mathcal{P}_c f"] \arrow[d, "e_{\mathbb{X}}"] & \mathcal{U}\mathcal{P}_c\mathbb{Y} \arrow[d, "e_{\mathbb{Y}}"'] \\
\mathcal{P}\mathcal{U}\mathbb{X} \arrow[r, "\mathcal{P}\mathcal{U}f"] & \mathcal{P}\mathcal{U}\mathbb{Y}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{U}\mathcal{P}_c\mathcal{P}_c\mathbb{X} \arrow[r, "\mathcal{U}\mathcal{P}_c\mathcal{P}_c f"] \arrow[d, "\mu_{\mathbb{X}}"] \arrow[dd, bend right, "\mu_X"'] & \mathcal{U}\mathcal{P}_c\mathcal{P}_c\mathbb{Y} \arrow[d, "\mu_{\mathbb{Y}}"'] \arrow[dd, bend left, "\mu_Y"] \\
\mathcal{U}\mathcal{P}_c\mathbb{X} \arrow[r, "\mathcal{U}\mathcal{P}_c f"] \arrow[d, "e_{\mathbb{X}}"] & \mathcal{U}\mathcal{P}_c\mathbb{Y} \arrow[d, "e_{\mathbb{Y}}"'] \\
\mathcal{P}\mathcal{U}\mathbb{X} \arrow[r, "\mathcal{P}\mathcal{U}f"] & \mathcal{P}\mathcal{U}\mathbb{Y}
\end{tikzcd}$$

However, the outer square of the diagram does commute - due to naturality of η (of $\mathcal{P}$), the lower square does commute - due to naturality of e, the outside triangles also do - due to the definitions of both η's and e, and e is injective. As a consequence, the upper square commutes as well.

For μ, notice that also $\mu_{\mathbb{X}}$ is a convex homomorphism from $\mathcal{P}_c\mathcal{P}_c\mathbb{X}$ to $\mathcal{P}_c\mathbb{X}$, and all the arguments that we used for naturality of η apply to the naturality of μ (of $\mathcal{P}_c$) as well, when looking at the right diagram above. So, μ is natural as well.

Clearly, η and μ (of $\mathcal{P}_c$) satisfy the compatibility conditions of the definition of a monad, as so do η and μ (of $\mathcal{P}$). ◀

Proof of Proposition 25. Since c is a convex algebra homomorphism, we have that for all $p \in (0,1)$ and $a \in L$, $c(px_1 + \bar{p}x_2)(a) = (pc(x_1) + \bar{p}c(x_2))(a)$. The latter is equivalent, by definition of $(-)^L$ (see Section 5.3), to $(pc(x_1)(a) + \bar{p}c(x_2)(a))$. If there is $i \in \{1,2\}$ such that $c(x_i)(a) = *$, then $(pc(x_1)(a) + \bar{p}c(x_2)(a)) = *$ (see Section 5.2). If not, then both $c(x_1)(a)$ and $c(x_2)(a)$ are in $\mathcal{P}_c\mathbb{S}$: $(pc(x_1)(a) + \bar{p}c(x_2)(a))$ is by definition (see Section 5.1) the set $\{py_1 + \bar{p}y_2 \mid y_1 \in c(x_1)(a) \text{ and } y_2 \in c(x_2)(a)\}$. ◀

B Proofs for Section 6

Proof of Lemma 27. First, given c, consider the map $\alpha \circ \mathcal{M}c$. We need to show that $\alpha \circ \mathcal{M}c$ is an algebra homomorphism from the free algebra $\mathcal{F}S$ to $H\mathcal{F}S$ in $\mathrm{EM}(\mathcal{M})$. This will show that $c^\# \colon \mathcal{F}S \to H\mathcal{F}S$ and $\mathcal{U}c^\# = \alpha \circ \mathcal{M}c$.

Note that

$$\left(\begin{array}{c} \mathcal{M}\mathcal{L}_1 S \\ \downarrow{\scriptstyle\alpha} \\ \mathcal{L}_1 S \end{array}\right) = \mathcal{H}\left(\begin{array}{c} \mathcal{M}\mathcal{M}S \\ \downarrow{\scriptstyle\mu} \\ \mathcal{M}S \end{array}\right).$$

The needed homomorphism property holds since the following diagram commutes:

$$\begin{tikzcd}
\mathcal{M}\mathcal{M}S \arrow[r, "\mathcal{M}\mathcal{M}c"] \arrow[d, "\mu"'] & \mathcal{M}\mathcal{M}\mathcal{L}_1 S \arrow[r, "\mathcal{M}\alpha"] \arrow[d, "\mu"] & \mathcal{M}\mathcal{L}_1 S \arrow[d, "\alpha"] \\
\mathcal{M}S \arrow[r, "\mathcal{M}c"] & \mathcal{M}\mathcal{L}_1 S \arrow[r, "\alpha"] & \mathcal{L}_1 S
\end{tikzcd}$$

as the left square commutes by the naturality of μ and the right one by the Eilenberg-Moore law for α.

Next, we show that the assignments $c \overset{\triangleright}{\mapsto} c^\#$ and $c^\# \overset{\triangleleft}{\mapsto} c$ are inverse to each other.

We have $\triangleleft \circ \triangleright(c) = c$ as $\alpha \circ \mathcal{M}c \circ \eta \overset{\text{nat.}\eta}{=} \alpha \circ \eta \circ c \overset{\text{EM}-\text{law}}{=} c$. Also, we have $\triangleright \circ \triangleleft(c^\#) = c^\#$ as

$$\begin{aligned}
\alpha \circ \mathcal{M}(\mathcal{U}c^\# \circ \eta) \quad &= \quad \alpha \circ \mathcal{M}\mathcal{U}c^\# \circ \mathcal{M}\eta \\
&\overset{(*)}{=} \quad \mathcal{U}c^\# \circ \mu \circ \mathcal{M}\eta \\
&\overset{(**)}{=} \quad \mathcal{U}c^\#
\end{aligned}$$

where the equality marked by $(*)$ holds since $\mathcal{U}c^\#$ is an algebra homomorphism, proven above, and the equality marked by $(**)$ holds by the monad law.

By the above, $\overline{\mathcal{F}}$ is well defined on objects. It remains to prove that for two $\mathcal{L}_1$-coalgebras on **Sets** (S, c_S) and (T, c_T), and a coalgebra homomorphism $h\colon (S, c_S) \to (T, c_T)$ with $c_T \circ h = \mathcal{L}_1 h \circ c_S$ we have that $\mathcal{M}h$ is a coalgebra homomorphism in $\mathrm{EM}(\mathcal{M})$ from $(\mathcal{F}S, c_S^\#)$ to $(\mathcal{F}T, c_T^\#)$.

We have

$$\begin{tikzcd}
\mathcal{U}\mathcal{F}S \arrow[r, "\mathcal{U}\mathcal{F}h"] \arrow[d, "\mathcal{U}\mathcal{F}c_S"] \arrow[dd, bend right=60, "\mathcal{U}c_S^\#"'] & \mathcal{U}\mathcal{F}T \arrow[d, "\mathcal{U}\mathcal{F}c_T"] \arrow[dd, bend left=60, "\mathcal{U}c_T^\#"] \\
\mathcal{U}\mathcal{F}\mathcal{L}_1 S \arrow[r, "\mathcal{U}\mathcal{F}\mathcal{L}_1 h"] \arrow[d, "\alpha_S"'] & \mathcal{U}\mathcal{F}\mathcal{L}_1 T \arrow[d, "\alpha_T"] \\
\mathcal{U}\mathcal{H}\mathcal{F}S \arrow[r, "\mathcal{L}_1 h"] & \mathcal{U}\mathcal{H}\mathcal{F}T
\end{tikzcd}$$

where the outer triangles commute by definition; the upper square commutes by assumption, i.e., since h is a homomorphism and $\mathcal{U}$ and $\mathcal{F}$ are functors; and the lower square simply states that $\mathcal{H}\mathcal{F}h$ is an arrow in $\mathrm{EM}(\mathcal{M})$ which of course holds as $\mathcal{H}$ and $\mathcal{F}$ are functors. ◀

Proof of Proposition 28. Assume that R is a congruence. Then $\mathcal{U}\mathbb{S}/R$ carries a $\mathcal{M}$-algebra, denoted by $\mathbb{S}/R$, and $[-]_R\colon \mathbb{S} \to \mathbb{S}/R$ is a map in $\mathrm{EM}(\mathcal{M})$. If R is a kernel bisimulation on $\overline{\mathcal{U}}(\mathbb{S}, c)$, then there exists a function $f\colon \mathcal{U}(\mathbb{S}/R) \to \mathcal{L}_2\mathcal{U}(\mathbb{S}/R)$ such that the outer square in the diagram below on the left commutes.

$$\begin{tikzcd}
\mathcal{U}\mathbb{S} \arrow[r, two heads, "\mathcal{U}[-]_R"] \arrow[d, "\mathcal{U}c"'] & \mathcal{U}(\mathbb{S}/R) \arrow[d, dotted, "c_R"'] \arrow[dd, bend left=40, "f"] \\
\mathcal{U}\mathcal{H}\mathbb{S} \arrow[r, "\mathcal{U}\mathcal{H}[-]_R"] \arrow[d, tail, "e_{\mathbb{S}}"'] & \mathcal{U}\mathcal{H}(\mathbb{S}/R) \arrow[d, tail, "e_{\mathbb{S}/R}"'] \\
\mathcal{L}_2\mathcal{U}\mathbb{S} \arrow[r, "\mathcal{L}_2\mathcal{U}[-]_R"'] & \mathcal{L}_2\mathcal{U}(\mathbb{S}/R)
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{U}\mathbb{S} \arrow[r, two heads, "\mathcal{U}[-]_R"] \arrow[d, "\mathcal{U}\mathcal{H}[-]_R \circ \mathcal{U}c"'] & \mathcal{U}(\mathbb{S}/R) \arrow[d, "f"] \arrow[dl, dotted, "c_R"'] \\
\mathcal{U}\mathcal{H}(\mathbb{S}/R) \arrow[r, tail, "e_{\mathbb{S}/R}"'] & \mathcal{L}_2\mathcal{U}(\mathbb{S}/R)
\end{tikzcd}$$

The bottom square commutes by naturality of e. The function c_R is obtained by the epi-mono factorisation structure on **Sets** as shown on the right. To conclude that R is a kernel bisimulation on $(\mathbb{S}, c)$, one only needs to show that c_R is a map in $\mathrm{EM}(\mathcal{M})$.

Let $\alpha \colon \mathcal{MU}\mathbb{S} \to \mathcal{U}\mathbb{S}$ denote the algebra structure of $\mathbb{S}$, that is $\mathbb{S} = (\mathcal{U}\mathbb{S}, \alpha)$. Similarly $\mathbb{S}/R = (\mathcal{U}(\mathbb{S}/R), \alpha_R)$, $\mathcal{H}\mathbb{S} = (\mathcal{UH}\mathbb{S}, \alpha_H)$ and $\mathcal{H}(\mathbb{S}/R) = (\mathcal{UH}(\mathbb{S}/R), \alpha_{RH})$.

$$\begin{tikzcd}
\mathcal{MU}\mathbb{S} \arrow[rr, two heads, "\mathcal{MU}[-]_R"] \arrow[dr, "\alpha"] \arrow[dd, "\mathcal{MU}c"'] & & \mathcal{MU}(\mathbb{S}/R) \arrow[dr, "\alpha_R"] \arrow[dd, "\mathcal{M}c_R"] & \\
& \mathcal{U}\mathbb{S} \arrow[rr, two heads, "\mathcal{U}[-]_R"] \arrow[dd, "\mathcal{U}c"'] & & \mathcal{U}(\mathbb{S}/R) \arrow[dd, "c_R"] \\
\mathcal{MUH}\mathbb{S} \arrow[rr, "\mathcal{MUH}[-]_R"'] \arrow[dr, "\alpha_H"'] & & \mathcal{MUH}(\mathbb{S}/R) \arrow[dr, "\alpha_{RH}"] & \\
& \mathcal{UH}\mathbb{S} \arrow[rr, "\mathcal{UH}[-]_R"'] & & \mathcal{UH}(\mathbb{S}/R)
\end{tikzcd}$$

Consider the above cube on **Sets**. The front face commutes by construction of c_R. The back face commutes as it is just $\mathcal{M}$ applied to the front face. The top face commutes because R is a congruence: the map $[-]_R \colon (\mathcal{U}\mathbb{S}, \alpha) \to (\mathcal{U}(\mathbb{S}/R), \alpha_R)$ is a $\mathcal{M}$-algebra homomorphism. Since $\mathcal{H}$ is a functor also $\mathcal{H}[-]_R \colon \mathcal{H}(\mathcal{U}\mathbb{S}, \alpha) \to \mathcal{H}(\mathcal{U}(\mathbb{S}/R), \alpha_R)$ is a $\mathcal{M}$-algebra homomorphism: this means that also the bottom face commutes. The leftmost face commutes since, by assumption, c is a homomorphism of $\mathcal{M}$-algebras.

To prove that also c_R is a homomorphism of $\mathcal{M}$-algebras amounts to checking that also the rightmost face commutes. For this it is essential that $\mathcal{M}$ preserves epis, as every **Sets**-endofunctor does, so that $\mathcal{MU}[-]_R$ is an epi. From this fact and the following derivation, we conclude that $c_R \circ \alpha_R = \alpha_{RH} \circ \mathcal{M}c_R$.

$$\begin{array}{rcl}
c_R \circ \alpha_R \circ \mathcal{MU}[-]_R & = & c_R \circ \mathcal{U}[-]_R \circ \alpha \\
& = & \mathcal{UH}[-]_R \circ \mathcal{U}c \circ \alpha \\
& = & \mathcal{UH}[-]_R \circ \alpha_H \circ \mathcal{MU}c \\
& = & \alpha_{RH} \circ \mathcal{MUH}[-]_R \circ \mathcal{MU}c \\
& = & \alpha_{RH} \circ \mathcal{M}c_R \circ \mathcal{MU}[-]_R
\end{array}$$

The other implication follows trivially from the fact that $\overline{\mathcal{U}}$ is a functor. ◀

Proof of Proposition 29. $\overline{\mathcal{U}}(\mathbb{S}, c)$ is a coalgebra for the functor $\mathcal{P}^L \colon \mathbf{Sets} \to \mathbf{Sets}$, namely a labeled transition system. It is well known that for these kind of coalgebras, behavioural equivalence ($\approx$) coincides with the standard notion of bisimilarity.

We can thus proceed by exploiting coinduction and prove that the following relation is a bisimulation (in the standard sense).

$$R = \{(p\zeta_1 + \bar{p}\xi_1, p\zeta_2 + \bar{p}\xi_2) \mid \zeta_1 \approx \zeta_2,\ \ \xi_1 \approx \xi_2 \text{ and } p \in [0,1]\}$$

Suppose that $p\zeta_1 + \bar{p}\xi_1 \overset{a}{\mapsto} \epsilon_1$. Then, by Proposition 25, $\epsilon_1 = p\zeta_1' + \bar{p}\xi_1'$ with $\zeta_1 \overset{a}{\mapsto} \zeta_1'$ and $\xi_1 \overset{a}{\mapsto} \xi_1'$. Since $\zeta_1 \approx \zeta_2$ and $\xi_1 \approx \xi_2$, there exists $\zeta_2' \approx \zeta_1'$ and $\xi_2' \approx \xi_1'$ such that $\zeta_2 \overset{a}{\mapsto} \zeta_2'$ and $\xi_2 \overset{a}{\mapsto} \xi_2'$. Again, by Proposition 25, $p\zeta_2 + \bar{p}\xi_2 \overset{a}{\mapsto} p\zeta_2' + \bar{p}\xi_2'$. Moreover, by definition of R, $p\zeta_1' + \bar{p}\xi_1' \mathrel{R} p\zeta_2' + \bar{p}\xi_2'$. ◀

Proof of Theorem 31. We have that $\overline{\mathcal{U}} \circ \overline{\mathcal{F}} \circ \mathcal{T}_{\text{conv}}(S, c_M) = (\mathcal{D}S, \hat{c}_M) = (\mathcal{D}S, (e_{\mathcal{F}S} + 1)^L \circ \mathcal{U}\bar{c}_M^\sharp)$. Since behavioural equivalence in $(\mathcal{D}S, (e_{\mathcal{F}S} + 1)^L \circ \mathcal{U}\bar{c}_M^\sharp)$ is bisimilarity for LTSs, by Corollary 30, all that we need to show is that $(\mathcal{D}S, \hat{c}_M) = (\mathcal{D}S, (e_{\mathcal{F}S} + 1)^L \circ \mathcal{U}\bar{c}_M^\sharp)$ is exactly the belief-state transformer M_{bs} induced by M, as announced by Table 1. Moreover, since $(e_{\mathcal{F}S} + 1)^L$ is the identity embedding, it suffices to understand $\mathcal{U}\bar{c}_M^\sharp$.

First, by definition, $\mathcal{T}_{\mathrm{conv}}(S, c_M) = (S, \bar{c}_M) = (S, \mathrm{conv}^L \circ c_M)$, and concretely,

$$s \xrightarrow{l}_c \xi \ \text{ in } \ M_c \ \text{ iff } \ \xi \in \bar{c}_M(s)(l) = \mathrm{conv}(c_M(s)(l)).$$

Then $\mathcal{U}\bar{c}_M^\sharp = \alpha \circ \mathcal{D}\bar{c}_M \colon \mathcal{D}S \to \mathcal{D}((\mathcal{C}S+1)^L) \to (\mathcal{C}S+1)^L$ by Lemma 27, where α is the algebra structure from Example 24. Concretely, we have, for a distribution[7] $[s_i \mapsto p_i]$

$$\begin{array}{rcl}\mathcal{U}\bar{c}_M^\sharp([s_i \mapsto p_i])(l) & = & \alpha \circ \mathcal{D}\bar{c}_M([s_i \mapsto p_i])(l) \\ & = & \alpha(\mathcal{D}\bar{c}_M([s_i \mapsto p_i])(l) \\ & = & \alpha([\bar{c}_M(s_i) \mapsto p_i])(l) \\ & = & (*)\end{array}$$

Now $[\bar{c}_M(s_i) \mapsto p_i]$ is the distribution that assigns probability p_i to the function $l \mapsto \{\xi_i \mid s_i \xrightarrow{l}_c \xi_i\}$ if $s_i \xrightarrow{l}_c$, and $l \mapsto *$ otherwise. Hence, by the convex operations corresponding to the algebra structure α from Example 24 we have

$$\begin{array}{rcl}(*) & = & \begin{cases} \sum p_i\{\xi_i \mid s_i \xrightarrow{l}_c \xi_i\}\ , & \forall i.\, s_i \xrightarrow{l}_c \\ *\ , & \exists i.\, s_i \not\xrightarrow{l}_c \end{cases} \\ & = & \begin{cases} \{\sum p_i\xi_i \mid s_i \xrightarrow{l}_c \xi_i\}\ , & \forall i.\, s_i \xrightarrow{l}_c \\ *\ , & \exists i.\, s_i \not\xrightarrow{l}_c \end{cases}\end{array}$$

which means[8] that $\sum p_i s_i = [s_i \mapsto p_i] \overset{l}{\mapsto} \sum p_i \xi_i$ in $(\mathcal{D}S, \hat{c}_M)$ whenever $s_i \xrightarrow{l}_c \xi_i$ which is exactly the same condition as $\sum p_i s_i \overset{l}{\mapsto} \sum p_i \xi_i$ in M_{bs}. This completes the proof that M_{bs} is exactly the coalgebra $(\mathcal{D}S, \hat{c}_M)$. ◀

C Proofs for Section 7

In this appendix we show the proofs for Propositions 33 and 34. While the first basically requires only Proposition 25, the second can be more elegantly illustrated by using the modular approach developed in [45] that we recall hereafter.

Up-to techniques can be combined in a number of interesting ways. For a map $\mathrm{f}\colon Rel_{\mathcal{D}(S)} \to Rel_{\mathcal{D}(S)}$, the n-iteration of f is defined as $\mathrm{f}^{n+1} = \mathrm{f} \circ \mathrm{f}^n$ and $\mathrm{f}^0 = \mathrm{id}$, the identity function. The omega iteration is defined as $\mathrm{f}^\omega(R) = \bigcup_{i=0}^\infty \mathrm{f}^i(R)$. The following result from [45] informs us that compatible up-to techniques can be composed resulting in other compatible techniques.

▶ **Lemma 38.** *The following functions are compatible:*

- id*: the identity function;*
- $\mathrm{f} \circ \mathrm{g}$*: the composition of compatible functions* f *and* g*;*
- $\bigcup F$*: the pointwise union of an arbitrary family F of compatible functions: $\bigcup F(R) = \bigcup_{\mathrm{f} \in F} \mathrm{f}(R)$;*
- f^ω*: the (omega) iteration of a compatible function* f*, defined as $\mathrm{f}^\omega(R) = \bigcup_{i=0}^\infty \mathrm{f}^i(R)$*

◀

Apart from conv, we are interested in the following up-to techniques.

[7] In order to avoid confusion between convex operations and distributions, for the moment we use this explicit notation for distributions.

[8] Coming back to the usual notation for distributions, as formal sums.

- the constant function r mapping every R into the identity relation $Id \subseteq \mathcal{D}(S) \times \mathcal{D}(S)$;
- the square function t mapping every R into $t(R) = \{(\zeta_1, \zeta_3) \mid \exists \zeta_2 \text{ s.t. } \zeta_1 R \zeta_2 R \zeta_3\}$;
- the opposite function s mapping every R into its opposite relation R^{-1}.

It is easy to check that all these functions are compatible. Lemma 38 allows us to combine them so to obtain novel compatible up-to techniques. For instance the equivalence closure $\mathrm{e}\colon Rel_{\mathcal{D}(S)} \to Rel_{\mathcal{D}(S)}$ can be decomposed as $(\mathrm{id} \cup \mathrm{r} \cup \mathrm{t} \cup \mathrm{s})^{\omega}$. The fact that e is compatible follows immediately from Lemma 38.

In a similar way, we can decompose cgr as $(\mathrm{id} \cup \mathrm{r} \cup \mathrm{t} \cup \mathrm{s} \cup \mathrm{conv})^{\omega}$. To prove that it is compatible, we have first to prove that conv is compatible.

Proof of Proposition 33. Assume that $(\epsilon_1, \epsilon_2) \in \mathrm{conv}(b(R))$. By definition of conv there exist $p \in [0, 1]$, $\zeta_1, \xi_1, \zeta_2, \xi_2 \in \mathcal{D}(S)$ such that

(a) $\epsilon_1 = p\zeta_1 + \bar{p}\xi_1$, $\epsilon_2 = p\zeta_2 + \bar{p}\xi_2$,

(b) $(\zeta_1, \zeta_2) \in \mathrm{b}(R)$ and $(\xi_1, \xi_2) \in \mathrm{b}(R)$.

To prove that $(\epsilon_1, \epsilon_2) \in \mathrm{b}(\mathrm{conv}(R))$, assume that $\epsilon_1 \overset{a}{\mapsto} \epsilon_1'$ for some $a \in A$ and $\epsilon_1' \in \mathcal{D}(S)$. Then, by (a) and Proposition 25, $\zeta_1 \overset{a}{\mapsto} \zeta_1'$ and $\xi_1 \overset{a}{\mapsto} \xi_1'$ and $\epsilon_1' = p\zeta_1' + \bar{p}\xi_1'$. By (b), there exist $\zeta_2', \xi_2' \in \mathcal{D}(S)$ such that

(c) $\zeta_2 \overset{a}{\mapsto} \zeta_2'$, $\xi_2 \overset{a}{\mapsto} \xi_2'$ and

(d) $(\zeta_1', \zeta_2') \in R$, $(\xi_1', \xi_2') \in R$.

From (c) and Proposition 25, it follows that $\epsilon_2 \overset{a}{\mapsto} p\zeta_2' + \bar{p}\xi_2'$. From (d), one obtains that $(\epsilon_1' = p\zeta_1' + \bar{p}\xi_1', p\zeta_2' + \bar{p}\xi_2') \in \mathrm{conv}(R)$.

One can proceed symmetrically for $\epsilon_2 \overset{a}{\mapsto} \epsilon_2'$.

Therefore, by definition of b, $(\epsilon_1, \epsilon_2) \in \mathrm{b}(\mathrm{conv}(R))$. ◀

Proof of Proposition 34. Observe that $\mathrm{cgr} = (\mathrm{id} \cup \mathrm{r} \cup \mathrm{t} \cup \mathrm{s} \cup \mathrm{conv})^{\omega}$. We know that r, s, t and conv (Proposition 33) are compatible. Compatibility of cgr follows by Lemma 38. ◀

D Detailed Introduction to Coalgebras

In this appendix we give a gentle introduction to (co)algebra that enables us to highlight the generic principles behind the semantics of probabilistic automata. The interested reader is referred to [30, 46, 33] for more details. We start by recalling the basic notions of category, functor and natural transformation, so that all of the results in the paper are accessible also to non-experts.

A category $\mathbf{C}$ is a collection of objects and a collection of arrows (or morphisms) from one object to another. For every object $X \in \mathbf{C}$, there is an identity arrow $id_X\colon X \to X$. For any three objects $X, Y, Z \in \mathbf{C}$, given two arrows $f\colon X \to Y$ and $g\colon Y \to Z$, there exists an arrow $g \circ f\colon X \to Z$. Arrow composition is associative and id_X is neutral w.r.t. composition. The standard example is **Sets**, the category of sets and functions.

A functor F from a category $\mathbf{C}$ to a category $\mathbf{D}$, notation $F\colon \mathbf{C} \to \mathbf{D}$, assigns to every object $X \in \mathbf{C}$, an object $FX \in \mathbf{D}$, and to every arrow $f\colon X \to Y$ in $\mathbf{C}$ an arrow $Ff\colon FX \to FY$ in $\mathbf{D}$ such that identity arrows and composition are preserved.

▶ **Example 39.** Examples of functors on **Sets** of particular interest to us are

1. The constant exponent functor $(-)^L$ for a set L, mapping a set X to the set X^L of all functions from L to X, and a function $f\colon X \to Y$ to $f^L\colon X^L \to Y^L$ with $f^A(g) = f \circ g$.
2. The termination functor $(-) + 1$ that maps a set X to the set $X + 1 = X \cup \{*\}$ with $* \notin X$, and a map $f\colon X \to Y$ to $f + 1\colon X + 1 \to Y + 1$ given by $f + 1(x) = f(x)$ for $x \in X$ and $f + 1(*) = *$.

3. The non-empty powerset functor $\mathcal{P}_{ne}$ mapping a set X to the set of its non-empty subsets $\mathcal{P}_{ne} = \{S \mid S \subseteq X \text{ and } S \neq \emptyset\}$ and a function $f\colon X \to Y$ to a $\mathcal{P}_{ne}f\colon \mathcal{P}_{ne}X \to \mathcal{P}_{ne}Y$ with $\mathcal{P}_{ne}f(S) = f(S)$ for $S \subseteq X$.
4. The powerset functor $\mathcal{P}$ mapping a set X to its powerset $\mathcal{P}X = \{S \mid S \subseteq X\}$ and on functions it is defined as the previous one. Observe that $\mathcal{P} = \mathcal{P}_{ne} + 1$.
5. The finitely supported probability distribution functor $\mathcal{D}$ is defined, for a set X and a function $f\colon X \to Y$, as

$$\mathcal{D}X \;=\; \{\varphi\colon X \to [0,1] \mid \sum_{x \in X} \varphi(x) = 1, \operatorname{supp}(\varphi) \text{ is finite}\} \qquad \mathcal{D}f(\varphi)(y) \;=\; \sum_{x \in f^{-1}(y)} \varphi(x).$$

   The support set of a distribution $\varphi \in \mathcal{D}X$ is defined as $\operatorname{supp}(\varphi) = \{x \in X \mid \varphi(x) \neq 0\}$.
6. The nonempty-convex-subsets-of-distributions monad [43, 29, 62] C maps a set X to the set of all nonempty convex subsets of distributions over X, and a function $f\colon X \to Y$ to the function $\mathcal{PD}f$.

A category $\mathbf{C}$ is concrete, if it admits a canonical forgetful functor $\mathcal{U}\colon \mathbf{C} \to \mathbf{Sets}$. By a forgetful functor we mean a functor that is identity on arrows. Intuitively, a concrete category has objects that are sets with some additional structure, e.g. algebras, and morphisms that are particular kind of functions.

Let $F\colon \mathbf{C} \to \mathbf{D}$ and $G\colon \mathbf{C} \to \mathbf{D}$ be two functors. A natural transformation $\sigma\colon F \Rightarrow G$ is a family of arrows $\sigma_X\colon FX \to GX$ in $\mathbf{D}$ such that the diagram on the right commutes for all arrows $f\colon X \to Y$.

$$\begin{tikzcd} FX \arrow[r, "Ff"] \arrow[d, "\sigma_X"] & FY \arrow[d, "\sigma_Y"] \\ GX \arrow[r, "Gf"] & GY \end{tikzcd}$$

Coalgebras provide an abstract framework for state-based system. Let $\mathbf{C}$ be a base category. A coalgebra is a pair (S, c) of a state space S (object in $\mathbf{C}$) and an arrow $c\colon S \to FS$ in $\mathbf{C}$ where $F\colon \mathbf{C} \to \mathbf{C}$ is a functor that specify the type of transitions. We will sometimes just say the coalgebra $c\colon S \to FS$, meaning the coalgebra (S, c). A coalgebra homomorphism from a coalgebra (S, c) to a coalgebra (T, d) is an arrow $h\colon S \to T$ in $\mathbf{C}$ that makes the diagram on the right commute. Coalgebras of a functor F and their coalgebra homomorphisms form a category that we denote by $\mathrm{Coalg}_{\mathbf{C}}(F)$.

$$\begin{tikzcd} S \arrow[r, "h"] \arrow[d, "c"] & T \arrow[d, "d"] \\ FS \arrow[r, "Fh"] & FT \end{tikzcd}$$

Coalgebras over a concrete category are equipped with a generic behavioural equivalence, which we define next. Let (S, c) be an F-coalgebra on a concrete category $\mathbf{C}$. An equivalence relation $R \subseteq \mathcal{U}S \times \mathcal{U}S$ is a kernel bisimulation (synonymously, a cocongruence) [56, 36, 67] if it is the kernel of a homomorphism, i.e., $R = \ker \mathcal{U}h = \{(s,t) \in \mathcal{U}S \times \mathcal{U}S \mid \mathcal{U}h(s) = \mathcal{U}h(t)\}$ for some coalgebra homomorphism $h\colon (S, c) \to (T, d)$ to some F-coalgebra (T, d). Two states s, t of a coalgebra are behaviourally equivalent notation $s \approx t$ iff there is a kernel bisimulation R with $(s, t) \in R$. A trivial but important property is that if there is a functor from one category of coalgebras (over a concrete category) to another, then this functor preserves behavioral equivalence: if two states are equivalent in a coalgebra of the first category, then they are also equivalent in the image under the functor in the second category.

We are now in position to connect probabilistic automata to coalgebras.

▶ **Proposition 40** ([4, 53]). *A probabilistic automaton $M = (S, L, \to)$ can be identified with a $(\mathcal{PD})^A$-coalgebra $c_M\colon S \to (\mathcal{PD}S)^A$ on* $\mathbf{Sets}$*, where $s \xrightarrow{a} \xi$ in M iff $\xi \in c_M(s)(a)$ in (S, c_M). Bisimilarity in M equals behavioural equivalence in (S, c_M), i.e., for two states $s, t \in S$ we have $s \sim t \Leftrightarrow s \approx t$.* ◀

It is also possible to provide convex bisimilarity semantics to probabilistic automata via coalgebraic behavioural equivalence, as the next proposition shows.

▶ **Proposition 41** ([43])**.** *Let $M = (S, L, \rightarrow)$ be a probabilistic automaton, and let $(S, \bar{c}_M)$ be a $(C+1)^A$-coalgebra on* **Sets** *defined by $\bar{c}_M(s)(a) = \mathrm{conv}(c_M(s)(a))$ where c_M is as before, if $c_M(s)(a) = \{\xi \mid s \stackrel{a}{\rightarrow} \xi\} \neq \emptyset$; and $\bar{c}_M(s)(a) = *$ if $c_M(s)(a) = \emptyset$. Convex bisimilarity in M equals behavioural equivalence in $(S, \bar{c}_M)$.* ◀

The connection between (S, c_M) and $(S, \bar{c}_M)$ in Proposition **??** is the same as the connection between M and M_c in Section 2. Abstractly, it can be explained using the following well known generic property.

▶ **Lemma 42** ([46, 4])**.** *Let $\sigma\colon F \Rightarrow G$ be a natural transformation from $F\colon \mathbf{C} \rightarrow \mathbf{C}$ to $G\colon \mathbf{C} \rightarrow \mathbf{C}$. Then $\mathcal{T}\colon \mathrm{Coalg}_{\mathbf{C}}(F) \rightarrow \mathrm{Coalg}_{\mathbf{C}}(G)$ given by*

$$\mathcal{T}(S \stackrel{c}{\rightarrow} FS) = (S \stackrel{c}{\rightarrow} FS \stackrel{\sigma_S}{\rightarrow} GS)$$

on objects and identity on morphisms is a functor. As a consequence, $\mathcal{T}$ preserves behavioural equivalence. If σ is injective, then $\mathcal{T}$ also reflects behavioural equivalence. ◀

▶ **Example 43.** We have that $\mathrm{conv}\colon \mathcal{PD} \Rightarrow C + 1$ given by $\mathrm{conv}(\emptyset) = *$ and $\mathrm{conv}(X)$ is the already-introduced convex hull for $X \subseteq \mathcal{D}S$, $X \neq \emptyset$ is a natural transformation. Therefore, $\mathrm{conv}^L\colon (\mathcal{PD})^L \Rightarrow (C+1)^L$ is one as well, defined pointwise. As a consequence from Lemma 11, we get a functor $\mathcal{T}_{\mathrm{conv}}\colon \mathrm{Coalg}_{\mathbf{Sets}}((\mathcal{PD})^L) \rightarrow \mathrm{Coalg}_{\mathbf{Sets}}((C+1)^L)$ and hence bisimilarity implies convex bisimilarity in probabilistic automata.

On the other hand, we have the injective natural transformation $\iota\colon C + 1 \Rightarrow \mathcal{PD}$ given by $\iota(X) = X$ and $\iota(*) = \emptyset$ and hence a natural transformation $\chi\colon (C+1)^L \Rightarrow (\mathcal{PD})^L$. As a consequence, convex bisimilarity coincides with strong bisimilarity on the "convex-closed" probabilistic automaton M_c, i.e., the coalgebra $(S, \bar{c}_M)$ whose transitions are all convex combinations of M-transitions.

D.1 Algebras for a Monad

The behaviour functor F often is, or involves, a monad $\mathcal{M}$, providing certain computational effects, such as partial, non-deterministic, or probabilistic computation.

More precisely, a monad is a functor $\mathcal{M}\colon \mathbf{C} \rightarrow \mathbf{C}$ together with two natural transformations: a unit $\eta\colon \mathrm{id}_{\mathbf{C}} \Rightarrow \mathcal{M}$ and multiplication $\mu\colon \mathcal{M}^2 \Rightarrow \mathcal{M}$. These are required to make the following diagrams commute, for $X \in \mathbf{C}$.

$$\begin{tikzcd}
\mathcal{M}X \arrow[r, "\eta_{\mathcal{M}X}"] \arrow[dr, equal] & \mathcal{M}^2X \arrow[d, "\mu_X"] & \mathcal{M}X \arrow[l, "\mathcal{M}\eta_X"'] \arrow[dl, equal] \\
& \mathcal{M}X &
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{M}^3X \arrow[r, "\mu_{\mathcal{M}X}"] \arrow[d, "\mathcal{M}\mu_X"'] & \mathcal{M}^2X \arrow[d, "\mu_X"] \\
\mathcal{M}^2 \arrow[r, "\mu_X"'] & \mathcal{M}X
\end{tikzcd}$$

We briefly describe two examples of monads on **Sets**.

- The unit of the powerset monad $\mathcal{P}$ is given by singleton $\eta(x) = \{x\}$ and multiplication by union $\mu(\{X_i \in \mathcal{P}X \mid i \in I\}) = \bigcup_{i \in I} X_i$.
- The unit of $\mathcal{D}$ is given by a Dirac distribution $\eta(x) = \delta_x = (x \mapsto 1)$ for $x \in X$ and the multiplication by $\mu(\Phi)(x) = \sum_{\varphi \in \mathrm{supp}(\Phi)} \Phi(\varphi) \cdot \varphi(x)$ for $\Phi \in \mathcal{DD}X$.

With a monad $\mathcal{M}$ on a category $\mathbf{C}$ one associates the Eilenberg-Moore category $\mathrm{EM}(\mathcal{M})$ of Eilenberg-Moore algebras. Objects of $\mathrm{EM}(\mathcal{M})$ are pairs $\mathbb{A} = (A, a)$ of an object $A \in \mathbf{C}$ and an arrow $a\colon \mathcal{M}A \to A$, making the first two diagrams below commute.

$$\begin{tikzcd} A \arrow[r, "\eta"] \arrow[dr, equal] & \mathcal{M}A \arrow[d, "a"] \\ & A \end{tikzcd} \qquad \begin{tikzcd} \mathcal{M}^2 A \arrow[r, "\mathcal{M}a"] \arrow[d, "\mu"'] & \mathcal{M}A \arrow[d, "a"] \\ \mathcal{M}A \arrow[r, "a"'] & A \end{tikzcd} \qquad \begin{tikzcd} \mathcal{M}A \arrow[r, "\mathcal{M}h"] \arrow[d, "a"'] & \mathcal{M}B \arrow[d, "b"] \\ A \arrow[r, "h"'] & B \end{tikzcd}$$

A homomorphism from an algebra $\mathbb{A} = (A, a)$ to an algebra $\mathbb{B} = (B, b)$ is a map $h\colon A \to B$ in $\mathbf{C}$ between the underlying objects making the diagram above on the right commute. The diagram in the middle thus says that the map a is a homomorphism from $(\mathcal{M}A, \mu_A)$ to $\mathbb{A}$. The forgetful functor $\mathcal{U}\colon \mathrm{EM}(\mathcal{M}) \to \mathbf{C}$ mapping an algebra to its carrier has a left adjoint $\mathcal{F}$, mapping an object $X \in \mathbf{C}$ to the (free) algebra $(\mathcal{M}X, \mu_X)$. We have that $\mathcal{M} = \mathcal{F} \circ \mathcal{U}$.

A category of Eilenberg-Moore algebras which is particularly relevant for our exposition is described in the following proposition. See [60] and [50] for the original result, but also [16, 17] or [28, Theorem 4] where a concrete and simple proof is given.

▶ **Proposition 44** ([60, 16, 17, 28]). *Eilenberg-Moore algebras of the finitely supported distribution monad $\mathcal{D}$ are exactly convex algebras as defined in Section 3. The arrows in the Eilenberg-Moore category $\mathrm{EM}(\mathcal{D})$ are convex algebra homomorphisms.* ◀

As a consequence, we will interchangeably use the abstract (Eilenberg-Moore algebra) and the concrete definition (convex algebra), whatever is more convenient. For the latter, we also just use binary convex operations, by Proposition 7, whenever more convenient.