The Power of Convex Algebras

Filippo Bonchi¹, Alexandra Silva², and Ana Sokolova³

¹ CNRS, ENS-Lyon, France
filippo.bonchi@ens-lyon.fr
² University College London, 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.

1998 ACM Subject Classification F.3 Logics and Meanings of Programs, G.3 Probability and Statistics, F.1.2 Modes of Computation, D.2.4 Software/Program verification

Keywords and phrases belief-state transformers, bisimulation up-to, coalgebra, convex algebra, convex powerset monad, probabilistic automata

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2017.23

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, 52]. Recent interest in research around semantics of probabilistic programming languages has led to new insights in connections between category theory, probability theory, and automata [59, 12, 27, 58, 44].

PA have been given various semantics, starting from strong bisimilarity [39], probabilistic (convex) bisimilarity [50, 49], to bisimilarity on distributions [18, 14, 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. [14, 15, 35, 1, 13, 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.

† This work has been partially supported by the project ANR-16-CE25-0011 REPAS, the ERC Starting Grant ProFoundNet (grant code 679127), and the FWF National Research Network RiSE/SHiNE project S11411-N23.
The theory of coalgebras [30, 46, 31] provides a toolbox for modelling and analysing different types of state machines. In a nutshell, a coalgebra is an arrow \( c : S \to FS \) for some functor \( F : C \to C \) on a category \( C \). 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 \( (\simeq) \), a coinductive proof technique and, for finite states machines, a procedure to check \( \approx \).

By tuning the parameters \( C \) and \( F \), one can retrieve many existing types of machines and their associated equivalences. For instance, by taking \( C = \text{Sets} \), the category of sets and functions, and \( FS = (\mathcal{P}S)^L \), the set of functions from \( L \) to subsets \( \mathcal{P} \) of probability distributions \( \mathcal{D} \) over \( S \), coalgebras \( c : S \to FS \) are in one-to-one correspondence with \( \text{PA} \) with labels in \( L \). Moreover, the associated notion of behavioural equivalence turns out to be the classical strong probabilistic bisimilarity of [39] (see [4, 54] 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 [50, 49].

In this paper, we take a coalgebraic outlook at the semantics of probabilistic automata as belief-state transformers: we wish to translate a \( \text{PA} \) into a labeled transition system on convex algebras. In turn, these can be transformed – without changing the underlying behavioural equivalence – into standard LTSs on \( \text{Sets} \). From an abstract perspective, both \( \mathcal{D} \) and \( \mathcal{P} \) are monads, hereafter denoted by \( \mathcal{M} \), and both \( \text{PA} \) and \( \text{NDA} \) can be regarded as coalgebras of type \( c : S \to FMS \).

In [53], a generalised determinisation transforming coalgebras \( c : S \to FMS \) into coalgebras \( \mathcal{M}S \to FMS \) was presented. This construction requires the existence of a lifting \( F \) of \( c \) to the category of algebras for the monad \( \mathcal{M} \). 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
Probabilistic automata are models of systems that involve both probability and nondeterminism. We start with their definition by Segala and Lynch [50].

**Definition 1.** A probabilistic automaton (PA) is a triple \( M = (S, L, \rightarrow) \) where \( S \) is a set of states, \( L \) is a set of actions or action labels, and \( \rightarrow \subseteq S \times L \times \mathcal{D}(S) \) is the transition relation. As usual, \( s \xrightarrow{a} \zeta \) stands for \((s, a, \zeta) \in \rightarrow\). ⊞

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 [50], and as transformers of belief states [10, 22, 13, 15, 14, 26] whose definitions we present next.

For the rest of the section, we fix a PA \( M = (S, L, \rightarrow) \).

**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 \( \xi, \xi' \in \mathcal{D}(S) \), that

\[
\begin{align*}
 s &\xrightarrow{a} \xi \Rightarrow \exists \xi' \in \mathcal{D}(S). \ t \xrightarrow{a} \xi' \wedge \xi \equiv_{R} \xi', \quad \text{and} \quad t \xrightarrow{a} \xi' \Rightarrow \exists \xi \in \mathcal{D}(S). \ s \xrightarrow{a} \xi \wedge \xi \equiv_{R} \xi'.
\end{align*}
\]

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. \( \sum_{s \in S} \nu(s, t) = \xi(s) \) 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 \((s, t) \in R\). ⊞
The Power of Convex Algebras

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

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

Here $\xrightarrow{c}$ denotes the convex transition relation, defined as follows: $s \xrightarrow{a} \xi$ 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 bisimilarity $R$ with $(s, t) \in R$.

Convex bisimilarity is (strong probabilistic) bisimilarity on the "convex closure" of the given PA. More precisely, consider the PA $M_c = (S, L, \xrightarrow{c})$ in which $s \xrightarrow{a} \xi$ whenever $s \in S$ and $\xi$ is in the convex hull (see Section 3 for a definition) of the set $\{ \xi \in \mathcal{D}(S) | s \xrightarrow{a} \xi \}$. 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, 13, 15, 14, 26, 11] to name a few, with behavioural equivalences on distributions. We were particularly inspired by the original work of Deng et al. [13, 15, 14] as well as [26]. Given a PA $M = (S, L, \rightarrow)$, consider the labeled transition system $M_{bs} = (\mathcal{D}S, L, \rightarrow)$ with states distributions over the original states of $M$, and transitions $\rightarrow \subseteq \mathcal{D}S \times L \times \mathcal{D}S$ defined by

$$\xi \xrightarrow{a} \zeta \text{ iff } \zeta = \sum_{i} p_i \xi_i, \ s_i \xrightarrow{a} \xi_i, \ \xi = \sum_{i} p_i \xi_i.$$  

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 $\xrightarrow{a} \xi$ if some of its support states can perform an $\xrightarrow{a}$ step. There are several proposed notions of equivalences on distributions [25, 18, 19, 22, 13, 10, 26] that mainly differ in the treatment of termination. See [26] for a detailed comparison.

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

Two distributions $\xi$ and $\zeta$ 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 $\delta_s$ denotes the Dirac distribution with $\delta_s(x) = 1$.

While the foundations of strong probabilistic bisimilarity are well-studied [54, 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 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

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

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 $p \in [0, 1]$ we write $p = 1 - p$.

Definition 5. A convex algebra $X$ is an algebra with signature $\mathcal{C}$, i.e., a set $X$ together with an operation $\sum_{i=0}^n p_i(-)$, 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: $\sum_{i=0}^n p_i \left( \sum_{j=0}^n q_{i,j} x_j \right) = \sum_{j=0}^n \left( \sum_{i=0}^n p_i q_{i,j} \right) x_j$.

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

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

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

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 $X_1$ and $X_2$ with operations $\sum_{i=0}^n p_i(-)$ and $\bigoplus_{i=0}^n p_i(-)$, respectively (and binary versions $+$ and $\oplus$, respectively) such that $px + \bar{py} = px \oplus \bar{py}$ for all $p, x, y$, then $X_1 = X_2$.

One can also see (1) as a definition, see e.g. [60, Definition 1]. We make the connection explicit with the next proposition, cf. [60, Lemma 1-Lemma 4].

Proposition 7. Let $X$ be a set with binary operations $px + \bar{py}$ for $x, y \in X$ and $p \in (0, 1)$. For $x, y, z \in X$ and $p, q \in (0, 1)$, assume
- Idempotence: $px + \bar{px} = x$,
- Parametric commutativity: $px + \bar{py} = \bar{py} + px$,
- Parametric associativity: $p(qx + \bar{py}) + \bar{pz} = pqx + \bar{pq} \left( \frac{pq}{pq} y + \frac{\bar{pq}}{\bar{pq}} z \right)$,

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 $X$ be a convex algebra, with carrier $X$ and $C \subseteq X$. $C$ is convex if it is the carrier of a subalgebra of $X$, i.e., if $px + \bar{py} \in C$ for $x, y \in C$ and $p \in (0, 1)$. The convex hull of a set $S \subseteq X$, denoted $\text{conv}(S)$, is the smallest convex set that contains $S$.

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

\[\text{CONCUR 2017}\]

\[\text{F. Bonchi, A. Silva, and A. Sokolova 23:5}\]
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 the full version that also includes basic categorical definitions and more details than what we do here.

Coalgebras provide an abstract framework for state-based systems. Let \( C \) be a base category. A coalgebra is a pair \((S, c)\) of a state space \( S \) (object in \( C \)) and an arrow \( c: S \to FS \) in \( C \) where \( F: C \to C \) is a functor that specifies the type of transitions. We will sometimes just say the coalgebra \( c: S \to FS \), meaning the coalgebra \((S, c)\). A coalgebra homomorphism from a coalgebra \((S, c)\) to a coalgebra \((T, d)\) is an arrow \( h: S \to T \) in \( C \) that makes the diagram on the right commute.

Coalgebras of a functor and their coalgebra homomorphisms form a category that we denote by \( \text{Coalg}_C(F) \). Examples of functors on \( \text{Sets} \) which are of 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 \( f: X \to Y \) to \( f^L: X^L \to Y^L \) with \( f^L(g) = f \circ g \).
2. The powerset functor \( \mathcal{P} \) mapping a set \( X \) to its powerset \( \mathcal{P}X = \{ S \mid S \subseteq X \} \) and on functions \( f: X \to Y \) given by \( \mathcal{P}f: \mathcal{P}X \to \mathcal{P}Y \), \( \mathcal{P}(f)(U) = \{ f(u) \mid u \in U \} \).
3. The finitely supported probability distribution functor \( D \) is defined, for a set \( X \) and a function \( f: X \to Y \), as \( DX = \{ \varphi: X \to [0, 1] \mid \sum_{x \in X} \varphi(x) = 1, \text{supp}(\varphi) \text{ is finite} \} \)
   \[ DF(\varphi)(y) = \sum_{x \in f^{-1}(y)} \varphi(x). \]
   The support set of a distribution \( \varphi \in DX \) is defined as \( \text{supp}(\varphi) = \{ x \in X \mid \varphi(x) \neq 0 \} \).
4. The functor \( C \) [43, 28, 63] maps a set \( X \) to the set of all nonempty convex subsets of \( X \), and a function \( f: X \to Y \) to the function \( \mathcal{P}Df \).

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 \( X + 1 = X \cup \{ \ast \} \) with \( \ast \notin X \) and every function \( f: X \to Y \) by \( f + 1(\ast) = \ast \) 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 \( C \), with \( U: C \to \text{Sets} \) being the forgetful functor. An equivalence relation \( R \subseteq US \times US \) is a kernel bisimulation (synonymously, a cocongruence) [57, 36, 67] if it is the kernel of a homomorphism, i.e., \( R = \ker Uh = \{ (s, t) \in US \times US \mid Uh(s) = Uh(t) \} \) for some coalgebra homomorphism \( h: (S, c) \to (T, d) \) 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 simple but important property is that if there is a functor from one category of coalgebras (over a concrete category) to another that preserves the state space and is identity on morphisms, 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, 54]). A probabilistic automaton \( M = (S, L, \to) \) can be identified with a \((\mathcal{P}D)^L\)-coalgebra \( c_M: S \to (\mathcal{P}DS)^L \) on \( \text{Sets} \), where \( s \xrightarrow{\xi} s' \in M \) 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 \( s \sim t \iff s \approx t \).

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 $(S, c_M)$ be a $\mathbb{C}(C + 1)^L$-coalgebra on $\mathbf{Sets}$ defined by $c_M(s)(a) = \text{conv}(c_M(s)(a))$ where $c_M$ is as before, if $c_M(s)(a) = \{x \mid s \xrightarrow{a} \xi\} \neq \emptyset$; and $c_M(s)(a) = *$ if $c_M(s)(a) = \emptyset$. Convex bisimilarity in $M$ equals behavioural equivalence in $(S, c_M)$.

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

Lemma 11 ([46, 4]). Let $\sigma: F \Rightarrow G$ be a natural transformation from $F: C \rightarrow C$ to $G: C \rightarrow C$. Then $T: \text{Coalg}_C(F) \rightarrow \text{Coalg}_C(G)$ given by $T(S \xrightarrow{\sigma} FS) = (S \xrightarrow{\sigma} FS \xrightarrow{\varepsilon} GS)$ on objects and identity on morphisms is a functor that preserves behavioural equivalence. If $\sigma$ is injective, then $T$ also reflects behavioural equivalence.

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

Also, an injective natural transformation $\iota: C + 1 \Rightarrow \mathcal{P}D$ is given by $\iota(X) = X$ and $\iota(*) = \emptyset$ yielding an injective $\chi: (C + 1)L \Rightarrow (\mathcal{P}D)L$. As a consequence, convex bisimilarity coincides with strong bisimilarity on the “convex-closed” probabilistic automaton $M_\varepsilon$, 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 $\mathcal{M}: C \rightarrow C$ together with two natural transformations: a unit $\eta: \text{id}_C \Rightarrow \mathcal{M}$ and multiplication $\mu: \mathcal{M}^2 \Rightarrow \mathcal{M}$ that satisfy the laws $\mu \circ \eta_M = \text{id} = \mu \circ M\eta$ and $\mu \circ \mu_M = \mu \circ M\mu$.

An example that will be pivotal for our exposition is the finitely supported distribution monad. The unit of $\mathcal{D}$ is given by a Dirac distribution $\eta(x) = \delta_x = (x \xrightarrow{a} 1)$ for $x \in X$ and the multiplication by $\mu(\Phi)(x) = \sum_{\varphi \in \text{supp}(\Phi)} \Phi(\varphi) \cdot \varphi(x)$ for $\Phi \in \mathcal{D}D_X$.

With a monad $\mathcal{M}$ on a category $C$ one associates the Eilenberg-Moore category $\text{EM}(\mathcal{M})$ of Eilenberg-Moore algebras. Objects of $\text{EM}(\mathcal{M})$ are pairs $A = (A, a)$ of an object $A \in C$ and an arrow $a: MA \rightarrow A$, satisfying $a \circ \eta = \text{id}$ and $a \circ M\eta = a \circ \mu$.

A homomorphism from an algebra $A = (A, a)$ to an algebra $B = (B, b)$ is a map $h: A \rightarrow B$ in $C$ between the underlying objects satisfying $h \circ a = b \circ Mh$.

A category of Eilenberg-Moore algebras which is particularly relevant for our exposition is described in the following proposition. See [61] and [51] for the original result, but also [16, 17] or [29, Theorem 4] where a concrete and simple proof is given.

Proposition 13 ([61, 16, 17, 29]). 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 $\text{EM}(\mathcal{D})$ 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 [53], which serves as source of inspiration for our work. A functor $\mathcal{F}: \text{EM}(\mathcal{M}) \rightarrow \text{EM}(\mathcal{M})$ is said to be a lifting of a functor $F: \mathcal{C} \rightarrow \mathcal{C}$ if and only if $\mathcal{U} \circ \mathcal{F} = F \circ \mathcal{U}$. Here, $\mathcal{U}$ is the forgetful functor $\text{EM}(\mathcal{M}) \rightarrow \mathcal{C}$ mapping an algebra to its carrier. It has a left adjoint $\mathcal{F}$, mapping an object $X \in \mathcal{C}$ to the (free) algebra $(\mathcal{M}X, \mu_X)$. We have that $\mathcal{M} = \mathcal{U} \circ \mathcal{F}$.

Whenever $F: \mathcal{C} \rightarrow \mathcal{C}$ has a lifting $\mathcal{F}: \text{EM}(\mathcal{M}) \rightarrow \text{EM}(\mathcal{M})$, one has the following functors between categories of coalgebras.

\[
\begin{array}{ccc}
\text{Coalg}_\mathcal{C}(F\mathcal{M}) & \xrightarrow{\mathcal{F}} & \text{Coalg}_\mathcal{C}((F\mathcal{M})) \\
\end{array}
\]

The functor $\mathcal{F}$ transforms every coalgebra $c: S \rightarrow F\mathcal{M}S$ over the base category into a coalgebra $\overset{\mathcal{F}}{c}: FS \rightarrow \mathcal{F}FS$. Note that this is a coalgebra on $\text{EM}(\mathcal{M})$: the state space carries an algebra, actually the freely generated one, and $\overset{\mathcal{F}}{c}$ 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 $\mathcal{U}$ simply forgets about the algebraic structure: $\overset{\mathcal{F}}{c}$ is mapped into

\[
\mathcal{U}\overset{\mathcal{F}}{c}: \mathcal{M}S = \mathcal{U}FS \rightarrow \mathcal{U}\mathcal{F}FS = F\mathcal{U}S = F\mathcal{M}S.
\]

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

By taking $F = 2 \times (-)^L$ and $\mathcal{M} = \mathcal{P}$, one transforms $c: S \rightarrow 2 \times (\mathcal{P}S)^L$ into $\mathcal{U}\overset{\mathcal{F}}{c}: \mathcal{P}S \rightarrow 2 \times (\mathcal{P}S)^L$. The former is a non-deterministic automaton (every $c$ of this type is a pairing $\langle o, t \rangle$ of $o: S \rightarrow 2$, defining the final states, and $t: S \rightarrow \mathcal{P}(S)^L$, defining the transition relation) and the latter is a deterministic automaton which has $\mathcal{P}S$ as states space. In [53], see also [32], it is shown that, for a certain choice of the lifting $\mathcal{F}$, 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 $c: S \rightarrow (\mathcal{P}DS)^L$ and belief state transformers are coalgebras of type $DS \rightarrow (\mathcal{P}DS)^L$. One would like to take $F = \mathcal{P}L$ and $\mathcal{M} = D$ and reuse the above construction but, unfortunately, $\mathcal{P}L$ does not have a suitable lifting to $\text{EM}(D)$. This is a consequence of two well known facts: the lack of a suitable distributive law $\rho: \mathcal{D} \mathcal{P} \Rightarrow \mathcal{P} \mathcal{D}$ [64] and the one-to-one correspondence between distributive laws and liftings, see e.g. [32]. In the next section, we will nevertheless provide a "powerset-like" functor on $\text{EM}(D)$ that we will exploit then in Section 6 to properly model PA as belief-state transformers.

---

\[\text{As shown in [64], there is no distributive law of the powerset monad over the distribution monad. Note that a "trivial" lifting and a corresponding distributive law of the powerset functor over the distribution monad exists, it is based on [11] and has been exploited in [32]. However, the corresponding "determinisation" is trivial, in the sense that its distribution bisimilarity coincides with bisimilarity, and it does not correspond to the belief-state transformer.}\]
5 Coalgebras on Convex Algebras

In this section we provide several functors on $\text{EM}(\mathcal{D})$ that will be used in the modelling of probabilistic automata as coalgebras over $\text{EM}(\mathcal{D})$. 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.

5.1 Convex Powerset on Convex Algebras

We now define a functor, the (nonempty) convex powerset functor, on $\text{EM}(\mathcal{D})$. Let $\mathbb{A}$ be a convex algebra. We define $\mathcal{P}_c \mathbb{A}$ to be $\mathcal{A}_c = (A_c, a_c)$ 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: $pC + \overline{p}D = \{ pc + \overline{pd} \mid c \in C, d \in D \}$. 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.

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. [48]. 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 $h: \mathbb{A} \to \mathbb{B}$, $\mathcal{P}_c h = \mathcal{P}h$. The following property ensures that we are on the right track.

► Proposition 14. $\mathcal{P}_c \mathbb{A}$ is a convex algebra. If $h: \mathbb{A} \to \mathbb{B}$ is a convex homomorphism, then so is $\mathcal{P}_c h: \mathcal{P}_c \mathbb{A} \to \mathcal{P}_c \mathbb{B}$. $\mathcal{P}_c$ is a functor on $\text{EM}(\mathcal{D})$. ◄

► Remark 15. $\mathcal{P}_c$ is not a lifting of $C$ to $\text{EM}(\mathcal{D})$, 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 $\mathcal{P}_\text{ne}$, the nonempty powerset functor, but we have an embedding natural transformation $e: \mathcal{U} \circ \mathcal{P}_c \Rightarrow \mathcal{P}_\text{ne} \circ \mathcal{U}$ given by $e(C) = C$, i.e., we are in the situation:

\[
\begin{array}{ccc}
\text{EM}(\mathcal{D}) & \xrightarrow{\mathcal{P}_c} & \text{EM}(\mathcal{D}) \\
\mathcal{F} & \Downarrow{\mu} & \Downarrow{\mu} \\
\text{Sets} & \xrightarrow{e} & \text{Sets} \\
\end{array}
\]

\[
\begin{array}{ccc}
\text{EM}(\mathcal{D}) & \xrightarrow{\mathcal{P}_\text{ne}} & \text{EM}(\mathcal{D}) \\
\mathcal{U} & \Downarrow{\mu} & \Downarrow{\mu} \\
\text{Sets} & \xrightarrow{e} & \text{Sets} \\
\end{array}
\]

The right diagram in Remark 15 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 16. $\mathcal{P}_c$ is a monad on $\text{EM}(\mathcal{D})$, with $\eta$ and $\mu$ 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 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 [56] 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 \( \mathbb{A} + \{*\} \) for \(* \notin \mathbb{A} \) and convex operations given by
\[
px \oplus \bar{py} = \begin{cases} 
px + \bar{py}, & x, y \in \mathbb{A}, \\
*, & x = * \text{ or } y = *.
\end{cases}
\] (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].

\[\text{Proposition 17 [56, 23].} \quad \mathbb{A} + 1 \text{ as defined above is a convex algebra that extends } \mathbb{A} \text{ by a single element. The map } h + 1 \text{ obtained with the termination functor in } \text{Sets} \text{ is a convex homomorphism if } h : \mathbb{A} \to \mathbb{B} \text{ is. The assignments } (-) + 1 \text{ give a functor on } \text{EM}(\mathcal{D}). \quad\]

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

\[\text{Lemma 18.} \quad \text{The functor } (-) + 1 \text{ is a lifting of the termination functor to } \text{EM}(\mathcal{D}). \quad\]

\[\text{Remark 19.} \quad \text{Note that we are abusing notation here: Our termination functor } (-) + 1 \text{ on } \text{EM}(\mathcal{D}) \text{ is not the coproduct } (-) + 1 \text{ in } \text{EM}(\mathcal{D}). \text{ The coproduct was concretely described in [33, Lemma 4], and the coproduct } X + 1 \text{ has a much larger carrier than } X + 1. \text{ Nevertheless, we use the same notation as it is very intuitive and due to Lemma 18.}\]

### 5.3 Constant Exponent on Convex Algebras

We now show the existence of a constant exponent functor on \(\text{EM}(\mathcal{D})\). Let \(L\) be a set of labels or actions. Let \(\mathbb{A}\) be a convex algebra. Consider \(\mathbb{A}^L\) with carrier \(\mathbb{A}^L = \{f \mid f : L \to \mathbb{A}\}\) and operations defined (pointwise) by \((pf + \bar{pg})(l) = pf(l) + \bar{pg}(l)\).

The following property follows directly from the definitions.

\[\text{Proposition 20.} \quad \mathbb{A}^L \text{ is a convex algebra. If } h : \mathbb{A} \to \mathbb{B} \text{ is a convex homomorphism, then so is } h^L : \mathbb{A}^L \to \mathbb{B}^L \text{ defined as in } \text{Sets}. \text{ Hence, } (-)^L \text{ defined above is a functor on } \text{EM}(\mathcal{D}). \quad\]

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

\[\text{Lemma 21.} \quad \text{The constant exponent } (-)^L \text{ on } \text{EM}(\mathcal{D}) \text{ is a lifting of the constant exponent functor } (-)^L \text{ on } \text{Sets}. \quad\]

\[\text{Example 22.} \quad \text{Consider a free algebra } \mathcal{F}S = (\mathcal{D}S, \mu) \text{ of distributions over the set } S. \text{ By applying first the functor } \mathcal{P}_c, \text{ then } (-) + 1 \text{ and then } (-)^L, \text{ one obtains the algebra}
\]

\[\left(\mathcal{P}_c \mathcal{F}S + 1\right)^L = \left(\frac{\mathcal{D} \left((CS + 1)^L\right)}{\alpha} \right)
\]

where \(CS\) is the set of non-empty convex subsets of distributions over \(S\), and \(\alpha\) corresponds to the convex operations\(^3\) \(\sum p_i f_i\) defined by

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

\(^3\) In this case, for future reference, it is convenient to spell out the \(n\)-ary convex operations.
5.4 Transition Systems on Convex Algebras

We now compose the three functors introduced above to properly model transition systems as coalgebras on $\EM(D)$. The functor that we are interested in is $\coalg_{\EM(M)}(\mathcal{H})$: $\EM(D) \to \EM(D)$. A coalgebra $(S, c)$ 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 $c: S \to (\mathcal{P}_c S + 1)^L$ is a homomorphism of convex algebras. This property entails compositionality: the transitions of a composite state $px_1 + \bar{px}_2$ are fully determined by the transitions of its components $x_1$ and $x_2$, as shown in the next proposition. We write $x \xrightarrow{a} y$ for $x, y \in S$, the carrier of $S$ if $y \in c(x)(a)$, and $x \not\xrightarrow{a}$ if $c(x)(a) = *$.

Proposition 23. Let $(S, c)$ be a $(\mathcal{P}_c + 1)^L$-coalgebra, and let $x_1, x_2, y_1, y_2, z$ be elements of $S$, the carrier of $S$. Then, for all $p \in (0, 1)$, and $a \in L$

- $px_1 + \bar{px}_2 \xrightarrow{a} z$ iff $z = py_1 + \bar{py}_2$, $x_1 \xrightarrow{a} y_1$ and $x_2 \xrightarrow{a} y_2$;
- $px_1 + \bar{px}_2 \not\xrightarrow{a}$ iff $x_1 \not\xrightarrow{a} y_1$ or $x_2 \not\xrightarrow{a} y_2$.

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 $(\mathcal{P}_c + 1)^L$-coalgebra and that, in the latter, behavioural equivalence coincides with the standard notion of bisimilarity for LTSs (Proposition 27).

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 15 how $\mathcal{P}_c$ is related to $C$ and $\mathcal{P}_{nc}$. The following definition is the obvious generalisation.

Definition 24. Let $\mathcal{M}: \text{Sets} \to \text{Sets}$ be a monad and $\mathcal{L}_1, \mathcal{L}_2: \text{Sets} \to \text{Sets}$ be two functors. A functor $\mathcal{H}: \EM(\mathcal{M}) \to \EM(\mathcal{M})$ 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 $e: \mathcal{U} \circ \mathcal{H} \Rightarrow \mathcal{L}_2 \circ \mathcal{U}$ 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 $\mathcal{L}_2$.

\[
\begin{array}{ccc}
\text{Sets} & \xrightarrow{\mathcal{M}} & \text{Sets} \\
\xrightarrow{\mathcal{L}_1} & \cong & \xleftarrow{\mathcal{L}_2}
\end{array}
\]

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

\[
\begin{array}{ccc}
\text{Coalg}_{\text{Sets}}(\mathcal{L}_1) & \xrightarrow{\mathcal{F}} & \text{Coalg}_{\EM(\mathcal{M})}(\mathcal{H}) \\
\xrightarrow{\mathcal{P}} & & \xleftarrow{\mathcal{P}}
\end{array}
\]

We first define $\mathcal{F}$. Take an $\mathcal{L}_1$-coalgebra $(S, c)$ and recall that $\mathcal{F}S$ is the free algebra $\mu: \mathcal{M}MMS \to \mathcal{M}S$. The left diagram in Definition 24 entails that $\mathcal{H}FS$ is an algebra $\alpha: \mathcal{M}\mathcal{L}_1 S \to \mathcal{L}_1 S$. We call $\mathcal{U}c^\sharp$ the composition $\mathcal{U}FS = \mathcal{M}S \xrightarrow{\mathcal{M}\mathcal{L}_1 S} \mathcal{L}_1 S = \mathcal{U}\mathcal{H}FS$. The next lemma shows that $c^\sharp: FS \to \mathcal{H}FS$ is a map in $\EM(M)$.
Lemma 25. There is a 1-1 correspondence between $L_1$-coalgebras on $\text{Sets}$ and $H$-coalgebras on $\text{EM}(\mathcal{M})$ with carriers free algebras:

$$c : S \rightarrow L_1S \quad \text{in} \quad \text{Sets}$$

$$c^\# : FS \rightarrow HFS \quad \text{in} \quad \text{EM}(\mathcal{M})$$

- given $c$, we have $Uc^\# = \alpha \circ Mc$ for $\alpha = HS$,
- given $c^\#$, we have $c = Uc^\# \circ \eta$.

The assignment $\mathcal{F}(S,c) = (FS,c^\#)$ and $\mathcal{F}(h) = Mh$ gives a functor $\mathcal{F} : \text{Coalg}_{\text{Sets}}(L_1) \rightarrow \text{Coalg}_{\text{EM}(\mathcal{M})}(H)$.

Now we can define $U : \text{Coalg}_{\text{EM}(\mathcal{M})}(H) \rightarrow \text{Coalg}_{\text{Sets}}(L_2)$ as mapping every coalgebra $(S,c)$ with $c : S \rightarrow HS$ to its $U$-coalgebra homomorphism $h : (S,c) \rightarrow (T,d)$ into $Uh = Uh$. Routine computations confirm that $U$ is a functor.

Since $U$ is a functor that keeps the state set constant and is identity on morphisms, every kernel bisimulation on $(S,c)$ is also a kernel bisimulation on $U(S,c)$. The converse is not true in general: a kernel bisimulation $R$ on $U(S,c)$ is a kernel bisimulation on $(S,c)$ only if it is a congruence with respect to the algebraic structure of $S$.

Formally, $R$ is a congruence if and only if the set $US/(S,c)$ of equivalence classes of $R$ carries an Eilenberg-Moore algebra and the function $U[-]_R : US \rightarrow US/R$ mapping every element of $US$ to its $R$-equivalence class is an algebra homomorphism.

Proposition 26. The following are equivalent:

- $R$ is a kernel bisimulation on $(S,c)$,
- $R$ is a congruence of $S$ and a kernel bisimulation of $U(S,c)$.

In particular, Proposition 26 and the following result ensure that the functor $U : \text{Coalg}_{\text{EM}(\mathcal{M})}(\mathcal{P}_c + 1)^L \rightarrow \text{Coalg}_{\text{Sets}}\mathcal{P}^L$ preserves and reflects $\approx$.

Proposition 27. Let $(S,c)$ be a $(\mathcal{P}_c + 1)^L$-coalgebra. Behavioural equivalence on $U(S,c)$ is a convex congruence. Hence, $U$ 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 $\mathcal{P}D^L$-coalgebra $(S,cM)$ on $\text{Sets}$, via the convex model $M_c$ obtained as $\mathcal{T}_{\text{conv}}(S,cM)$, to the belief state transformer $M_{bs}$. The latter coincides with $U \circ \mathcal{F} \circ \mathcal{T}_{\text{conv}}(S,cM)$.

Theorem 28. Let $(S,cM)$ be a probabilistic automaton. For all $\xi, \zeta \in D_{\mathcal{S}}$,

$$\xi \sim_{bs} \zeta \quad \Leftrightarrow \quad \xi \approx \zeta \quad \text{in} \quad U \circ \mathcal{F} \circ \mathcal{T}_{\text{conv}}(S,cM).$$

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

---

Convex congruences are congruences of convex algebras, see e.g. [55]. They are convex equivalences, i.e., closed under componentwise-defined convex combinations.
Indeed, all the distributions depicted above have infinitely many possible choices for \( i \). 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 $f: \text{Rel}_{\mathcal{D}(S)} \to \text{Rel}_{\mathcal{D}(S)}$, while a bisimulation up-to $f$ is a relation $R$ such that $R \subseteq b f(R)$. An up-to technique $f$ is said to be sound if, for all $R \in \text{Rel}_{\mathcal{D}(S)}$, $R \subseteq b f(R)$ entails that $R \subseteq \sim_d$. It is said to be compatible if $f b(R) \subseteq b f(R)$. In [45], it is shown that every compatible up-to technique is also sound.

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

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

**Proposition 30.** $\text{conv}$ is compatible.  

This result has two consequences: First, $\text{conv}$ is sound\(^5\) and thus one can prove $\sim_d$ by means of bisimulation up-to $\text{conv}$; Second, $\text{conv}$ can be effectively combined with other compatible up-to techniques (for more details see [45] or the full version). In particular, by combining $\text{conv}$ with up-to equivalence – which is well known to be compatible – one obtains up-to congruence $\text{cgr}: \text{Rel}_{\mathcal{D}(S)} \to \text{Rel}_{\mathcal{D}(S)}$. This technique maps a relation $R$ into its congruence closure: the smallest relation containing $R$ which is a congruence.

**Proposition 31.** $\text{cgr}$ is compatible.

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

**Example 32.** 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 $R = \{(x_2, y_2), (x_3, y_1), (x_1, \frac{1}{2}y_1 + \frac{1}{2}y_2), (x_0, y_0)\}$ is a bisimulation up-to $\text{cgr}$: consider $(x_1, \frac{1}{2}y_1 + \frac{1}{2}y_2)$ (the other pairs are trivial) and observe that

\[
\begin{align*}
  x_1 \xrightarrow{a} & \frac{1}{2}x_1 + \frac{1}{2}x_2 \\
  \xrightarrow{b} & \frac{1}{2}x_3 + \frac{1}{2}x_2 \\
  \text{cgr}(R) \xrightarrow{a} & \frac{1}{2}y_1 + \frac{1}{2}y_2 \\
  \xrightarrow{b} & \frac{1}{2}y_3 + \frac{1}{2}y_2
\end{align*}
\]

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 $\text{cgr}(R)$. In symbols, if $x_1 \xrightarrow{\zeta} \xi = p(\frac{1}{2}x_1 + \frac{1}{2}x_2) + \bar{p}(\frac{1}{2}x_3 + \frac{1}{2}x_2)$, then $\frac{1}{2}y_1 + \frac{1}{2}y_2 \xrightarrow{a} \xi = p(\frac{1}{2}y_1 + \frac{1}{2}y_2) + \bar{p}(\frac{1}{2}y_3 + \frac{1}{2}y_2)$ and $(\zeta, \xi) \in \text{cgr}(R)$.

Recall that in Example 29, 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 32 is a finite bisimulation up-to $\text{cgr}$. It turns out that one can always check $\sim_d$ by means of only finite bisimulations up-to. The key to this result is the following theorem, recently proved in [55].

**Theorem 33.** Congruences of finitely generated convex algebras are 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 27). In other words there exists a finite relation $R$ such that $\text{cgr}(R) = \sim_d$. Such $R$ is a finite bisimulation up-to $\text{cgr}$:

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

**Corollary 34.** Let $(S, L, \rightarrow)$ 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 $\text{cgr}$ $R$ such that $(\zeta_1, \zeta_2) \in R$.  

---

\(^5\) In [47] a similar up-to technique called “up-to lifting” is defined in the context of probabilistic $\lambda$-calculus and proven sound.
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 34 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 NDA. 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 [62, 3, 34]. Despite the fact that our structures are coalgebras on algebras, they are not bialgebras: but still ≈ is a congruence and it is amenable to up-to techniques. We believe that lax bialgebra may provide some deeper insights.

References

23:16 The Power of Convex Algebras


The Power of Convex Algebras


