The Theory of Traces for Systems with Nondeterminism and Probability

Filippo Bonchi
University of Pisa, Italy

Ana Sokolova
University of Salzburg, Austria

Valeria Vignudelli
Univ Lyon, CNRS, ENS Lyon, UCB Lyon 1, LIP, France

***Abstract*—This paper studies trace-based equivalences for systems combining nondeterministic and probabilistic choices. We show how trace semantics for such processes can be recovered by instantiating a coalgebraic construction known as the generalised powerset construction. We characterise and compare the resulting semantics to known definitions of trace equivalences appearing in the literature. Most of our results are based on the exciting interplay between monads and their presentations via algebraic theories.**

1. Introduction

Systems exhibiting both nondeterministic and probabilistic behaviour are abundantly used in verification [1], [2], [3], [4], [5], [6], [7], AI [8], [9], [10], and studied from semantics perspective [11], [12], [13]. Probability is needed to quantitatively model uncertainty and belief, whereas nondeterminism enables modelling of incomplete information, unknown environment, implementation freedom, or concurrency. At the same time, the interplay of nondeterminism and probability has been posing some remarkable challenges [14], [15], [16], [17], [18], [19], [20], [21].

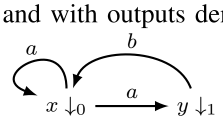
Figure 1 shows a nondeterministic probabilistic system (NPLTS) that we use as a running example.

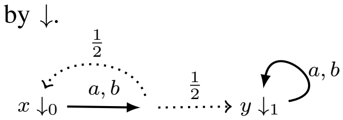
Traces and trace semantics [22] for nondeterministic probabilistic systems have been studied for several decades within concurrency theory and AI using resolutions or schedulers—entities that resolve the nondeterminism. Most proposals of trace semantics in the literature [23], [24], [25], [26] are based on such auxiliary notions of resolutions and differ on how these resolutions are defined and combined. We call such approaches *local-view* approaches.

On the other hand, the theory of coalgebra [27], [28] provides uniform generic approaches to trace semantics of various kinds of systems and automata, via Kleisli traces [29] or generalised determinisation [30], providing e.g. an abstract treatment of language equivalence for automata. We use the term *global-view* approaches for the coalgebraic methods via generalised determinisation.

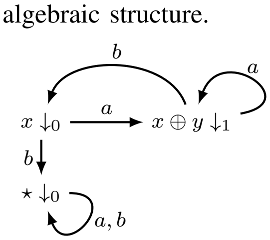
In this paper, we propose a theory of trace semantics for nondeterministic probabilistic systems that unifies the local and the global view. We start by taking the global-view approach founded on algebras and coalgebras and inspired by automata theory, and study determinisation of NPLTS in this framework. Then we find a way to mimic the local-view approach and show that we can recover known trace semantics from the literature. We introduce now the main pieces of our puzzle, and show how everything combines together in the theory of traces for NPLTS.

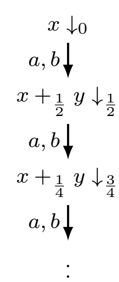
In order to illustrate our approach, it is convenient to recall nondeterministic automata (NDA) and Rabin probabilistic automata (PA) [31]. Both NDA and PA can be described as maps $\langle o, t\rangle\colon X \to O \times (MX)^A$ where X is a set of states, A is the set of labels, $o\colon X \to O$ is the output function assigning to each state in X an observation, and $t\colon X \to (MX)^A$ is the transition function that assigns to each state x in X and to each letter a of the alphabet A an element of MX that describes the choice of a next state. For NDA, this is a nondeterministic choice; for PA, the choice is governed by a probability distribution. An NDA state observes one of two possible values which qualify the state as accepting or not. A state in a PA observes a real number in $[0, 1]$. Below we depict an example NDA (on the left) and an example PA (on the right) with labels $A = \{a, b\}$ and with outputs denoted by $\downarrow$.





The type of choice, modelled abstractly by a monad M, is often linked to a concrete algebraic theory, the presentation of M. Having such a presentation is a valuable tool, since it provides a finite syntax for describing finite branching. For nondeterministic choice this is the algebraic theory of semilattices (with bottom), for probabilistic choice it is the algebraic theory of convex algebras. Once we have such an algebraic presentation, we have a determinised automaton (as depicted below) and we inductively compute the output value after executing a trace by following the algebraic structure.





This work has been supported by the European Research Council (ERC) under the European Union's Horizon 2020 programme (CoVeCe, grant agreement No 678157), by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR), and by the Austrian FWF National Research Network RiSE/SHiNE S11411-N23.

Here $x \oplus y$ denotes the nondeterministic choice of x or y, and $x +_p y$ the probabilistic choice where x is chosen with probability p and y with probability $1-p$. For example, in the determinised PA we have, since $x \xrightarrow{a} x +_{\frac{1}{2}} y$ and $y \xrightarrow{a} y$:

$$x +_{\frac{1}{2}} y \xrightarrow{a} (x +_{\frac{1}{2}} y) +_{\frac{1}{2}} y = x +_{\frac{1}{4}} y$$

and hence the output of $x +_{\frac{1}{4}} y$ is $o(x) +_{\frac{1}{4}} o(y) = \frac{3}{4}$ giving us the probability of x executing the trace aa. Our computation is enabled by having the right algebraic structure on the set of observations: a semilattice on $\{0,1\}$ and a convex algebra on $[0,1]$. The induced semantics is language equivalence and probabilistic language equivalence, respectively.

This is the approach of trace semantics via a determinisation [30], founded in the abstract understanding of automata as coalgebras and computational effects as monads.

We develop a theory of traces for NPLTS using such approach. For this purpose we take the monad for nondeterminism and probability [17] with origins in [14], [18], [19], [20], [21], [32], namely, the monad C of nonempty convex subsets of distributions, and provide all necessary and convenient infrastructure for generalised determinisation. The necessary part is having an algebra of observations, the convenient part is giving an algebraic presentation in terms of *convex semilattices*. These are algebras that are at the same time a semilattice and a convex algebra, with a distributivity axiom distributing probability over nondeterminism. Having the presentation we can write, for example

$$x \xrightarrow{a} x_1 \oplus (x_3 +_{\frac{1}{2}} x_2)$$

for the NPLTS from Figure 1.

The presentation for C is somewhat known, although not explicitly proven, in the community — proving it and putting it to good use is part of our contribution which, in our opinion, drastically clarifies and simplifies the trace theory of systems with nondeterminism and probability.

Remarkably, necessity and convenience go hand in hand on this journey. Having the presentation enables us to clearly identify what are the *interesting* algebras necessary for describing trace and testing semantics (with tests being finite traces). We identify three different algebraic theories: the *theory of pointed convex semilattices*, the *theory of convex semilattices with bottom*, and the *theory of convex semilattices with top*. These theories give rise to three interesting semantics by taking as algebras of observations those freely generated by a singleton set. We prove their concrete characterisations: the free convex semilattice with bottom is carried by $[0,1]$ with $\max$ as semilattice operation and standard convex algebra operations; the free convex semilattice with top is carried by $[0,1]$ with $\min$ as semilattice operation; and the pointed convex semilattice freely generated by 1 is carried by the set of closed intervals in $[0,1]$ where the semilattice operation combines two intervals by taking their minimum and their maximum, and the convex operations are given by Minkowski sum.

We call the resulting three semantics *may trace*, *must trace* and *may-must trace semantics* since there is a close correspondence with *probabilistic testing semantics* [33], [34], [35], [36] when tests are taken to be just the finite

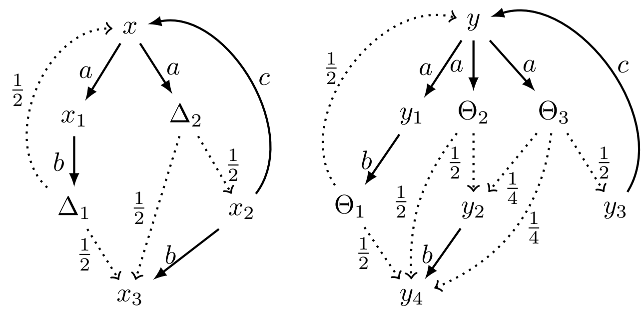


Figure 1. NPLTS

traces in A^*. Indeed, the may trace semantics gives the greatest probability with which a state passes a given test; the must trace semantics gives the smallest probability with which a state passes a given test, and the may-must trace semantics gives the closed interval ranging from the smallest to the greatest.

From the abstract theory, we additionally get that:

1) The induced equivalence can be proved coinductively by means of proof-techniques known as *bisimulations up-to* [37]. More precisely, it holds that up-to $\oplus$ and up-to $+_p$ are compatible [38] techniques.
2) The equivalence is implied by the standard branching-time equivalences for NPLTS, namely bisimilarity and convex bisimilarity [7], [39].
3) The equivalence is *backward compatible* w.r.t. trace equivalence for LTS and for reactive probabilistic systems (RPLTS): When regarding an LTS and RPLTS as a nondeterministic probabilistic system, standard trace equivalence coincides with our may trace equivalence and with our three semantics, respectively.

Last but certainly not least, we show that the global view coincides with the local one, namely that our three semantics can be elegantly characterised in terms of resolutions. The may-trace semantics assigns to each trace the greatest probability with which the trace can be performed, with respect to any resolution of the system; the must-trace semantics assigns the smallest one. It is important to remark here that our resolutions differ from those previously proposed in the literature in the fact that they are reactive rather than fully probabilistic. We observe that however this difference does not affect the greatest probability, and we can therefore show that the may-trace coincides with the randomized $\sqcup$-trace equivalence in [25], [26], [40].

Synopsis. We recall monads and algebraic theories in Section 2. We provide a presentation for the monad C in Section 3 (Theorem 4) and combine it with termination in Section 4. We then recall, in Section 5, the generalised determinisation and show an additional useful result (Theorem 16). All these pieces are put together in Section 6, where we introduce our three semantics and discuss their properties. The correspondence of the global view with the local one is illustrated in Section 7 (Theorem 23). An extended version of the paper containing all the proofs, additional examples and a proper treatment of the bisimulation up-to techniques can be found in [41].

2. Monads and Algebraic Theories

In this paper, on the algebraic side, we deal with Eilenberg-Moore algebras of a monad on the category $\mathbf{Sets}$ of sets and functions, for which we also give presentations in terms of operations and equations, i.e., algebraic theories.

2.1. Monads

A monad on $\mathbf{Sets}$ is a functor $M\colon \mathbf{Sets} \to \mathbf{Sets}$ together with two natural transformations: a unit $\eta\colon \mathrm{Id} \Rightarrow M$ and multiplication $\mu\colon M^2 \Rightarrow M$ that satisfy the laws $\mu \circ \eta M = \mu \circ M\eta = \mathit{id}$ and $\mu \circ M\mu = \mu \circ \mu M$.

We next introduce several monads on $\mathbf{Sets}$, relevant to this paper. Each monad can be seen as giving side-effects.

Nondeterminism. The finite powerset monad $\mathcal{P}$ maps a set X to its finite powerset $\mathcal{P}X = \{U \mid U \subseteq X,\ U \text{ is finite}\}$ and a function $f\colon X \to Y$ to $\mathcal{P}f\colon \mathcal{P}X \to \mathcal{P}Y$, $\mathcal{P}f(U) = \{f(u) \mid u \in U\}$. The unit η of $\mathcal{P}$ is given by singleton, i.e., $\eta(x) = \{x\}$ and the multiplication μ is given by union, i.e., $\mu(S) = \bigcup_{U \in S} U$ for $S \in \mathcal{P}\mathcal{P}X$. Of particular interest to us in this paper is the submonad $\mathcal{P}_{ne}$ of non-empty finite subsets, that acts on functions just like the (finite) powerset monad, and has the same unit and multiplication. We rarely mention the unrestricted (not necessarily finite) powerset monad, which we denote by $\mathcal{P}_u$. We sometimes write $\overline{f}$ for $\mathcal{P}_u f$ in this paper.

Probability. The finitely supported probability distribution monad $\mathcal{D}$ is defined, for a set X and a function $f\colon X \to Y$, as

$$\mathcal{D}X = \{\varphi\colon X \to [0,1] \mid \sum_{x \in X} \varphi(x) = 1,\ \mathrm{supp}(\varphi) \text{ is finite}\}$$

$$\mathcal{D}f(\varphi)(y) = \sum_{x \in f^{-1}(y)} \varphi(x).$$

The support set of a distribution $\varphi \in \mathcal{D}X$ is $\mathrm{supp}(\varphi) = \{x \in X \mid \varphi(x) \neq 0\}$. 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 $\Phi \in \mathcal{D}\mathcal{D}X$. We sometimes write $\sum_{i \in I} p_i x_i$ for a distribution φ with $\mathrm{supp}(\varphi) = \{x_i \mid i \in I\}$ and $\varphi(x_i) = p_i$.

Termination. The termination monad, also called lift and denoted by $\cdot + 1$ maps a set X to the set $X + 1$, where $+$ denotes the coproduct in $\mathbf{Sets}$, which amounts to disjoint union, and $1 = \{\star\}$. For a coproduct $A + B$ we write $\mathit{in}_l\colon A \to A + B$ and $\mathit{in}_r\colon B \to A + B$ for the left and right coproduct injections, respectively. This monad maps a function $f\colon X \to Y$ to the function $f + 1\colon X + 1 \to Y + 1$ defined, as expected, by $(f + 1)(\mathit{in}_l(x)) = \mathit{in}_l(f(x))$ for $x \in X$ and $(f + 1)(\mathit{in}_r(\star)) = \mathit{in}_r(\star)$. The unit of the termination monad is given by the left injection, $\eta\colon X \to X+1$ with $\eta(x) = \mathit{in}_l(x)$ and the multiplication by $\mu(\mathit{in}_l \circ \mathit{in}_l(x)) = \mathit{in}_l(x)$ for $x \in X$, $\mu(\mathit{in}_l \circ \mathit{in}_r(\star)) = \mathit{in}_r(\star)$, and $\mu(\mathit{in}_r(\star)) = \mathit{in}_r(\star)$. If clear from the context, we may omit explicit mentioning of the injections, and write for example $(f + 1)(x) = x$ for $x \in X$ and $(f + 1)(\star) = \star$.

2.2. Monad Maps, Quotients and Submonads

A monad map from a monad M to a monad $\hat{M}$ is a natural transformation $\sigma\colon M \Rightarrow \hat{M}$ that makes the following diagrams commute, with η, μ and $\hat{\eta}, \hat{\mu}$ denoting the unit and multiplication of M and $\hat{M}$, respectively, and $\sigma\sigma = \sigma \circ M\sigma = \hat{M}\sigma \circ \sigma_M$.

$$\begin{tikzcd} X \arrow[r, "\eta"] \arrow[dr, "\hat{\eta}"'] & MX \arrow[d, "\sigma"] \\ & \hat{M}X \end{tikzcd} \qquad \begin{tikzcd} MMX \arrow[r, "\sigma\sigma"] \arrow[d, "\mu"'] & \hat{M}\hat{M}X \arrow[d, "\hat{\mu}"] \\ MX \arrow[r, "\sigma"'] & \hat{M}X \end{tikzcd}$$

If $\sigma\colon MX \to \hat{M}X$ is an epi monad map, then $\hat{M}$ is a *quotient* of M. If it is a mono, then M is a *submonad* of $\hat{M}$. If it is an iso, the two monads are isomorphic.

2.3. Distributive Laws

Let (M, η, μ) and $(\hat{M}, \hat{\eta}, \hat{\mu})$ be two monads. A *monad distributive law* of M over $\hat{M}$ is a natural transformation $\lambda\colon M\hat{M} \Rightarrow \hat{M}M$ that commutes appropriately with the units and the multiplications of the monads, see [41] Appendix C.

Given a monad distributive law $\lambda\colon M\hat{M} \Rightarrow \hat{M}M$, we get a composite monad $\bar{M} = \hat{M}M$ with unit $\bar{\eta} = \hat{\eta}\eta$ and multiplication $\bar{\mu} = \hat{\mu}\mu \circ \hat{M}\lambda M$.

For any monad M on $\mathbf{Sets}$, there exists a distributive law $\iota\colon M + 1 \Rightarrow M(\cdot + 1)$ defined as

$$\iota_X = \big(\ MX + 1 \xrightarrow{[M\mathit{in}_l, \eta_{X+1} \circ \mathit{in}_r]} M(X + 1)\ \big). \quad (1)$$

As a consequence, $M(\cdot + 1)$ is a monad. Moreover, we get the following useful property.

Lemma 1. *Whenever $\sigma\colon M \Rightarrow \hat{M}$ is a monad map, also $\sigma(\cdot+1)\colon M(\cdot+1) \Rightarrow \hat{M}(\cdot+1)$ is a monad map. Injectivity of σ implies injectivity of $\sigma(\cdot + 1)$.* ☐

2.4. Algebraic Theories

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

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

A homomorphism from an algebra $\mathbb{A} = (A, a)$ to an algebra $\mathbb{B} = (B, b)$ is a map $h\colon A \to B$ between the underlying sets making the third diagram above commute.

In this paper we care for both categorical algebra, algebras of a monad, and their presentations in terms of algebraic theories and their models. An algebraic theory is a pair (Σ, E) of signature Σ (a set of operation symbols) and a set of equations E (a set of pairs of terms). A (Σ, E)-algebra, or a model of the algebraic theory (Σ, E) is an algebra $\mathbb{A} = (A, \Sigma_A)$ with carrier set A and a set of operations

Σ_A, one for each operation symbol in Σ, that satisfies the equations in E. A homomorphism from a (Σ, E)-algebra $\mathbb{A} = (A, \Sigma_A)$ to a (Σ, E)-algebra $\mathbb{B} = (B, \Sigma_B)$ is a function $h\colon A \to B$ that commutes with the operations, i.e., $h \circ f_A = f_B \circ h^n$ for all n-ary $f \in \Sigma$, and f_A, f_B its interpretations in $\mathbb{A}, \mathbb{B}$, respectively. (Σ, E)-algebras together with their homomorphisms form a category and a variety.

Definition 2. *A* presentation *of a monad M is an algebraic theory, (Σ, E) such that the category (variety) of (Σ, E)-algebras is isomorphic to* $\mathrm{EM}(M)$.

Given a presentation (Σ, E) of a monad M, M is isomorphic to the monad $M_{\Sigma,E}$ of Σ-terms modulo E-equations, i.e., there is an isomorphism monad map between them. Given a signature Σ, the free monad $T_\Sigma = T_{\Sigma,\emptyset}$ of terms over Σ maps a set X to the set of all Σ-terms with variables in X, and $f\colon X \to Y$ to the function that maps a term over X to a term over Y obtained by substitution according to f. The unit maps a variable X to itself, and the multiplication is term composition. We have that $T_{\Sigma,E}$ is a quotient of T_Σ. Moreover, for two sets of equations $E_1 \subseteq E_2$ we have that the monad T_{Σ,E_2} is a quotient of T_{Σ,E_1}. In the sequel we present several algebraic theories that give presentations to the monads of interest.

Presenting the monad $\mathcal{P}_{ne}$. Let Σ_N be the signature consisting of a binary operation $\oplus$. Let E_N be the following set of axioms.

$$\begin{array}{ccc} (x \oplus y) \oplus z & \overset{(A)}{=} & x \oplus (y \oplus z) \\ x \oplus y & \overset{(C)}{=} & y \oplus x \\ x \oplus x & \overset{(I)}{=} & x \end{array}$$

The algebraic theory (Σ_N, E_N) of *semilattices* provides a presentation for the monad $\mathcal{P}_{ne}$. We refer to this theory as the theory of nondeterminism. To avoid confusion later, it is convenient to fix here the interpretation of $\oplus$ as a join (rather than a meet) and, thus, to think of the induced order as $x \sqsubseteq y$ iff $x \oplus y = y$.

Presenting the monad $\mathcal{D}$. Let Σ_P be the signature consisting of binary operations $+_p$ for all $p \in (0, 1)$. Let E_P be the following set of axioms.

$$\begin{array}{ccc} (x +_q y) +_p z & \overset{(A_p)}{=} & x +_{pq} (y +_{\frac{p(1-q)}{1-pq}} z) \\ x +_p y & \overset{(C_p)}{=} & y +_{1-p} x \\ x +_p x & \overset{(I_p)}{=} & x \end{array}$$

Here, (A_p), (C_p), and (I_p) are the axioms of parametric associativity, commutativity, and idempotence. The algebraic theory (Σ_P, E_P) of *convex algebras*, see [42], [43], [44], [45], [46], provides a presentation for the monad $\mathcal{D}$.

Another presentation of convex algebras is given by the algebraic theory with infinitely many operations denoting arbitrary (and not only binary) convex combinations — see [41] Appendix C for more details. This allows us to interchangeably use binary convex combinations or arbitrary convex combinations whenever more convenient. Moreover, we can write binary convex combinations $+_p$ for $p \in [0, 1]$ and not just $p \in (0, 1)$. We refer to the theory of convex algebras as the algebraic theory for probability.

Presenting $\cdot + 1$. The algebraic theory (Σ_T, E_T) for the termination monad consists of a single constant (nullary operation symbol) $\Sigma_T = \{\star\}$ and no equations $E_T = \emptyset$. This is called the theory of *pointed sets*.

Combining Algebraic Theories. Algebraic theories can be combined in a number of general ways: by taking their coproduct, their tensor, or by means of distributive laws (see e.g. [47]). Unfortunately, these abstract constructions do not lead to a presentation for the monad we are interested in. We will thus devote the next section to show a "hand-made" presentation for this monad.

We conclude this section with a well known fact that can be easily proved, for instance by taking the distributive law in (1): given a presentation (Σ, E) for a monad M, the monad $M(\cdot + 1)$ is presented by the theory (Σ', E) where Σ' is Σ together with an extra constant $\star$. For instance, the subdistributions monad $\mathcal{D}(\cdot + 1)$ is presented by the theory $(\Sigma_P \cup \Sigma_T, E_P)$ of *pointed convex algebras*, also known as *positive convex algebras*. The theory $(\Sigma_N \cup \Sigma_T, E_N)$ of *pointed semilattices* provides instead a presentation for the monad $\mathcal{P}_{ne}(\cdot + 1)$. It is interesting to observe that the powerset monad $\mathcal{P}$ is presented by adding to $(\Sigma_N \cup \Sigma_T, E_N)$ the equation

$$x \oplus \star \overset{(B)}{=} x$$

leading to the theory of *semilattices with bottom*. The theory of *semilattices with top* can be obtained by adding instead the following equation:

$$x \oplus \star \overset{(T)}{=} \star.$$

Similar axioms can be added to the theory of pointed convex algebras $(\Sigma_P \cup \Sigma_T, E_P)$. The axiom

$$x +_p \star \overset{(B_p)}{=} x$$

makes the probabilistic structure collapse, see Figure 6 in [41] Appendix C for the details. On the other hand, the axiom

$$x +_p \star \overset{(T_p)}{=} \star$$

quotients the monad $\mathcal{D}(\cdot + 1)$ into $\mathcal{D} + 1$: intuitively, each term of this theory is either a sum of only variables (a distribution) or an extra element ($\star$). This axiom describes the unique functorial way of adding termination to a convex algebra, the so-called black-hole behaviour of $\star$, cf. [48].

3. Algebraic Theory for Nondeterminism and Probability

In this section we recall the definition of the monad C for probability and nondeterminism, give its presentation via *convex semilattices*, and present examples of C-algebras.

3.1. The monad C of convex subsets of distributions

The monad C origins in the field of domain theory [19], [20], [21], and in the work of Varacca and Winskel [14], [18], [32]. Jacobs [17] gives a detailed study of (a generalisation of) this monad.

For a set X, CX is the set of non-empty, finitely-generated convex subsets of distributions on X, i.e.,

$$CX = \{S \subseteq \mathcal{D}X \mid S \neq \emptyset, \mathrm{conv}(S) = S,\\ S \text{ is finitely generated}\}.$$

Recall that, for a subset S of a convex algebra, $\mathrm{conv}(S)$ is the convex closure of S, i.e., the smallest convex set that contains S, i.e.,

$$\mathrm{conv}(S) = \{\sum p_i x_i \mid p_i \in [0,1], \sum p_i = 1, x_i \in S\}.$$

We say that a convex set S is generated by its subset B if $S = \mathrm{conv}(B)$. In such a case we also say that B is a basis for S. A convex set S is finitely generated if it has a finite basis.

For a function $f\colon X \to Y$, $Cf\colon CX \to CY$ is given by

$$Cf(S) = \{\mathcal{D}f(d) \mid d \in S\} = \overline{\mathcal{D}f}(S).$$

The unit of C is $\eta\colon X \to CX$ given by $\eta(x) = \{\delta_x\}$.

The multiplication of C, $\mu\colon CCX \to CX$ can be expressed in concrete terms as follows [17]. Given $S \in CCX$,

$$\mu(S) = \bigcup_{\Phi \in S} \{ \sum_{U \in \mathrm{supp}\,\Phi} \Phi(U) \cdot d \mid d \in U\}.$$

3.2. The presentation of C

We now introduce the algebraic theory (Σ_{NP}, E_{NP}) of *convex semilattices*, that gives us the presentation of C and thus provides an algebraic theory for nondeterminism and probability.

A convex semilattice $\mathbb{A}$ is an algebra $\mathbb{A} = (A, \oplus, +_p)$ with a binary operation $\oplus$ and for each $p \in (0,1)$ a binary operation $+_p$ satisfying the axioms $(A), (C), (I)$ of a semilattice, the axioms $(A_p), (C_p), (I_p)$ for a convex algebra, and the following distributivity axiom:

$$(x \oplus y) +_p z \overset{(D)}{=} (x +_p z) \oplus (y +_p z)$$

Hence, (Σ_{NP}, E_{NP}) is given by $\Sigma_{NP} = \Sigma_N \cup \Sigma_P$ and $E_{NP} = E_N \cup E_P \cup \{(D)\}$.

In every convex semilattice there also holds a convexity law, of which we directly present the generalised version in the following lemma.

Lemma 3. *Let $\mathbb{A} = (A, \oplus, +_p)$ be a convex semilattice. Then for all $n \in \mathbb{N}$, all $a_1, \ldots a_n \in A$ and all $p_1, \ldots, p_n \in [0,1]$ with $\sum_{i=1}^n p_i = 1$ we have*

$$a_1 \oplus \ldots \oplus a_n \oplus \sum_{i=1}^{n} p_i a_i \overset{(C)}{=} a_1 \oplus \ldots \oplus a_n. \quad \square$$

For $p \in [0,1]$ we set $\overline{p} = 1 - p$. Let X be an arbitrary set. We define Σ_{NP}-operations on CX by

$$S_1 \oplus S_2 = \mathrm{conv}(S_1 \cup S_2)$$

and for $p \in (0,1)$

$$S_1 +_p S_2 = \{\varphi \mid \varphi = p\varphi_1 + \overline{p}\varphi_2 \text{ for some } \varphi_1 \in S_1, \varphi_2 \in S_2\}$$

where $p\varphi_1 + \overline{p}\varphi_2 = \varphi_1 +_p \varphi_2$ is the binary convex combination of φ_1 and φ_2 in $\mathcal{D}X$, defined point-wise. Note that $S_1 +_p S_2$ is the Minkowski sum of two convex sets. If convenient, we may sometimes also write, as usual, $pS_1 + \overline{p}S_2$ for the Minkowski sum $S_1 +_p S_2$.

To prove the presentation theorem, we identify a generic proof method that we show in [41] Appendix D for lack of space. We encourage the reader to read it, also for many other useful properties that deepen the understanding of convex semilattices.

Theorem 4. *The theory for nondeterminism and probability (Σ_{NP}, E_{NP}), i.e., the theory of convex semilattices, is a presentation for the monad C.* $\square$

Remark 5. *Theorem 4 is to some extent known*[1] *but we could not find a proof of it in the literature. In [14], [18] a monad for probability and nondeterminism is given starting from a similar algebraic theory (with somewhat different basic algebraic structure). There is also another possible way of combining probability with nondeterminism, by distributing $\oplus$ over $+_p$ (see e.g. [15], [49]).*

Remark 6. *Having the presentation enables us to identify and interchangeably use convex subsets of distributions and terms in Σ_{NP} modulo equations in E_{NP}. This is particularly useful in examples and our further developments. Note that in the syntactic view $\eta(x)$ is identified with the term x.*

The presentation is a valuable tool in many situations where reasoning with algebraic theories is more convenient than reasoning with monads. For instance, it is much easier to check whether a certain algebra is a (Σ_{NP}, E_{NP})-model, than to check that it is an algebra for the monad C. We illustrate this with three (Σ_{NP}, E_{NP}) models that play a key role in our further results and exposition.

The max convex semilattice. $\mathbb{M}\mathrm{ax} = ([0,1], \max, +_\mathrm{p})$ is a (Σ_{NP}, E_{NP})-algebra when taking $\oplus$ to be $\max\colon [0,1] \times [0,1] \to [0,1]$ and $+_p$ the standard convex combination $+_p\colon [0,1] \times [0,1] \to [0,1]$ with $x +_p y = p \cdot x + \overline{p} \cdot y$ for $x, y \in [0,1]$. To check that this is a (Σ_{NP}, E_{NP}) model, it is enough to prove that $\max$ satisfies the axioms in E_N, that $+_p$ satisfies the axioms in E_P, and that they satisfy the axiom (D), namely that $\max(x,y) +_p z = \max(x +_p z, y +_p z)$.

The min convex semilattice. $\mathbb{M}\mathrm{in} = ([0,1], \min, +_\mathrm{p})$ is obtained similarly by taking $\oplus$ to be $\min\colon [0,1] \times [0,1] \to [0,1]$ rather than $\max$, and gives another example of a (Σ_{NP}, E_{NP})-algebra. It is indeed very simple to check that $([0,1], \min)$ forms a semilattice and that the distributivity law holds.

1. Personal communication with Gordon Plotkin.

The min-max interval convex semilattice. We consider the algebraic structure $\mathbb{M}_{\mathcal{I}} = (\mathcal{I}, \text{min-max}, +_p^{\mathcal{I}})$ for $\mathcal{I}$ the set of intervals on $[0,1]$, i.e.,

$$\mathcal{I} = \{[x,y] \mid x, y \in [0,1] \text{ and } x \leq y\}.$$

For $[x_1, y_1], [x_2, y_2] \in \mathcal{I}$, we define $\text{min-max}\colon \mathcal{I} \times \mathcal{I} \to \mathcal{I}$ as

$$\text{min-max}([x_1, y_1], [x_2, y_2]) = [\min(x_1, x_2), \max(y_1, y_2)]$$

and $+_p^{\mathcal{I}}\colon \mathcal{I} \times \mathcal{I} \to \mathcal{I}$ by

$$[x_1, y_1] +_p^{\mathcal{I}} [x_2, y_2] = [x_1 +_p x_2,\ y_1 +_p y_2].$$

The fact that this is a model for (Σ_{NP}, E_{NP}) follows easily from the fact that $\mathbb{M}\text{ax}$ and $\mathbb{M}\text{in}$ are models for (Σ_{NP}, E_{NP}).

Remark 7. *The fact that* $\mathbb{M}\text{ax}$ *and* $\mathbb{M}\text{in}$ *are* C*-algebras on* $[0,1]$ *was already proven in [50], without an algebraic presentation. Having the algebraic presentation significantly simplifies the proofs.*

4. Adding termination

So far, we have provided a presentation for the monad C which combines probability and nondeterminism. In order to properly model NPLTS, we need a last ingredient: termination. As discussed in Section 2, termination is given by the monad $\cdot + 1$ which can always be safely combined with any monad. Following the discussion at the end of Section 2, the theory $\mathcal{PCS} = (\Sigma_{NP} \cup \Sigma_T, E_{NP})$ presents the monad $C(\cdot + 1)$ which is the monad of finitely generated non empty convex sets of *sub*distributions.

We call this theory $\mathcal{PCS}$ since algebras for this theory are *pointed convex semilattices*, namely convex semilattices with a pointed element denoted by $\star$. A noteworthy example is $\mathbb{M}_{\mathcal{I},[0,0]} = (\mathcal{I}, \text{min-max}, +_p^{\mathcal{I}}, [0,0])$ where $\mathbb{M}_{\mathcal{I}} = (\mathcal{I}, \text{min-max}, +_p^{\mathcal{I}})$ is the convex semilattice of intervals from Section 3 and $[0,0]$ is the pointed element. Moreover, this is not just any pointed convex semilattice:

Proposition 8. $\mathbb{M}_{\mathcal{I},[0,0]} = (\mathcal{I}, \text{min-max}, +_p^{\mathcal{I}}, [0,0])$ *is the free pointed convex semilattice generated by a singleton set.*

Like for the monad $\mathcal{P}_{ne}$, there exist more than one interesting way of combining C with $\cdot + 1$. Rather than pointed convex semilattices, one can consider *convex semilattices with bottom*, namely algebras for the theory $\mathcal{CSB} = (\Sigma_{NP} \cup \Sigma_T, E_{NP} \cup \{(B)\})$ obtained by adding (B) to $\mathcal{PCS}$. Otherwise, one can add the axiom (T) and obtain the theory $\mathcal{CST} = (\Sigma_{NP} \cup \Sigma_T, E_{NP} \cup \{(T)\})$ of *convex semilattices with top*. We denote by $T_{\mathcal{CSB}}$ and $T_{\mathcal{CST}}$ the corresponding monads.

As we will illustrate in Section 5, particularly relevant for defining trace semantics is the free algebra $\mu\colon MM\{\bullet\} \to M\{\bullet\}$ generated by a singleton $\{\bullet\}$. In the next two propositions we identify these algebras for the monads $T_{\mathcal{CSB}}$ and $T_{\mathcal{CST}}$ in concrete terms.

Proposition 9. $\mathbb{M}\text{ax}_B = ([0,1], \max, +_p, 0)$ *is the free convex semilattice with bottom generated by* $1 = \{\bullet\}$. $\square$

Proposition 10. $\mathbb{M}\text{in}_T = ([0,1], \min, +_p, 0)$ *is the free convex semilattice with top generated by* $1 = \{\bullet\}$. $\square$

At this point the reader may wonder what happens when one considers the axioms (B_p) and (T_p) in place of (B) and (T). We have already shown at the end of Section 2.4, that the axiom (B_p) makes the probabilistic structure collapse. When focussing on the free algebra generated by $\{\bullet\}$, also quotienting by (T_p) is not really interesting: one can show by induction on the terms in $T_{\Sigma_{NP} \cup \Sigma_T}(\{\bullet\})$ that every term is equal via $E_{NP} \cup \{(T_p)\}$ to either $\bullet$ or $\star$ or $\bullet \oplus \star$.

So, we have found three interesting ways of combining termination with probability and nondeterminism. Table 1 summarises these theories, their monads, and their algebras.

5. Coalgebras and Determinisation

In this section, we briefly introduce coalgebra and the generalised determinisation [30] construction, as well as trace semantics by determinisation. We present some simple properties and a new important result concerning the semantics.

5.1. Coalgebra

The theory of coalgebra provides an abstract framework for state-based transition systems and automata. A *coalgebra* in $\mathbf{Sets}$ is a pair (S, c) of a state space S and a function $c\colon S \to FS$ where $F\colon \mathbf{Sets} \to \mathbf{Sets}$ is a functor that specifies the type of transitions. Sometimes we 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 a function $h\colon S \to T$ that satisfies $d \circ h = Fh \circ c$. Coalgebras of a functor F and their coalgebra homomorphisms form a category, denoted by $\text{Coalg}(F)$.

The final object in $\text{Coalg}(F)$, when it exists, is the *final* F*-coalgebra*. We write $\zeta\colon Z \xrightarrow{\cong} FZ$ for the final F-coalgebra. For every coalgebra $c\colon S \to FS$, there is a unique homomorphism $[\![\cdot]\!]_c$ to the final one, the *final coalgebra map*, making the diagram below commute:

$$\begin{tikzcd} FS \arrow[r, dashed, "F[\![\cdot]\!]_c"] & FZ \\ S \arrow[u, "c"] \arrow[r, dashed, "\exists!\,[\![\cdot]\!]_c"] & Z \arrow[u, "\cong"', "\zeta"] \end{tikzcd}$$

The *final coalgebra semantics* $\sim$ is the kernel of the final coalgebra map, i.e., two states s and t are equivalent in the final coalgebra semantics iff $[\![s]\!]_c = [\![t]\!]_c$.

Even without a final coalgebra, coalgebras over a concrete category are equipped with a generic behavioural equivalence. Let (S, c) be an F-coalgebra on $\mathbf{Sets}$. An equivalence relation $R \subseteq S \times S$ is a kernel bisimulation (synonymously, a cocongruence) [51], [52], [53] if it is the kernel of a homomorphism, i.e., $R = \ker h = \{(s,t) \in S \times S \mid h(s) = h(t)\}$ 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 $(s,t) \in R$.

Theory (Σ, E)	Monad M	free algebra $\mu_1 \colon MM1 \to M1$
$\mathcal{PCS} = (\Sigma_{NP} \cup \Sigma_T, E_{NP})$	$C(\cdot + 1) = T_{\mathcal{PCS}}$	$\mathbb{M}_{\mathcal{I},[0,0]} = (\mathcal{I}, \text{min-max}, +_p^{\mathcal{I}}, [0,0])$
$\mathcal{CSB} = (\Sigma_{NP} \cup \Sigma_T, E_{NP} \cup \{(B)\})$	$T_{\mathcal{CSB}}$	$\mathbb{M}\mathrm{ax}_B = ([0,1], \max, +_p, 0)$
$\mathcal{CST} = (\Sigma_{NP} \cup \Sigma_T, E_{NP} \cup \{(T)\})$	$T_{\mathcal{CST}}$	$\mathbb{M}\mathrm{in}_T = ([0,1], \min, +_p, 0)$

TABLE 1. THE THEORIES OF POINTED CONVEX SEMILATTICES, WITH BOTTOM, AND WITH TOP.

If a final coalgebra exists, then the behavioural equivalence and the final coalgebra semantics coincide, i.e., $\approx\, = \,\sim$.

The following are well-known examples of F-coalgebras on **Sets**:

1) Labelled transition systems, LTS, are coalgebras for the functor $F = (\mathcal{P}(\cdot))^A$. Behavioural equivalence coincides with strong bisimilarity.
2) Nondeterministic automata, NA, are coalgebras for $F = 2 \times (\mathcal{P}(\cdot))^A$ where $2 = \{0, 1\}$ is needed to differentiate whether a state is accepting or not.
3) Deterministic automata, DA, are coalgebras for $F = 2 \times (\cdot)^A$. The final coalgebra is carried by the set of all languages 2^{A^*}.
4) Moore automata, MA, are a slight generalisation of deterministic automata with observations O: they are coalgebras for $F = O \times (\cdot)^A$. The final coalgebra is carried by the set of all O-valued languages O^{A^*}.

Systems and Automata with M-effects. In general, for a monad M, we call an M^A-coalgebra a *system with M-effects*, and we call an $O \times M^A$-coalgebra an *automaton with M-effects and observations in O*. We write $c = \langle o, t \rangle$ for an automaton with M-effects and observations in O, where $o \colon X \to O$ is the observation map assigning observations to states, and $t \colon X \to (MX)^A$ is the transition structure.

For instance, an LTS is a system with $\mathcal{P}$-effects, and a nondeterministic automaton is an automaton with $\mathcal{P}$-effects and observations in 2. We now introduce the systems and automata that we focus on in this paper.

NPLTS. Nondeterministic probabilistic labelled transition systems, NPLTS, also known as simple Segala systems, are coalgebras for the functor $F = (\mathcal{PD}(\cdot))^A$. Behavioural equivalence coincides with strong probabilistic bisimilarity [54], [55]. Special cases of NPLTS are LTS, when all distributions are Dirac distributions, and reactive probabilistic labelled transition systems (RPLTS), when all subsets are at most singletons. An RPLTS is a coalgebra of the functor $(\mathcal{D}(\cdot) + 1)^A$.

Convex NPLTS. Convex NPLTS are coalgebras for $(C + 1)^A$. Behavioural equivalence coincides with convex probabilistic bisimilarity [16]. The move from NPLTS to convex NPLTS is given by a natural transformation $\mathrm{conv} \colon \mathcal{PD} \Rightarrow C + 1$ with $\mathrm{conv}(X)$ the convex hull for $X \subseteq \mathcal{D}S$, $X \neq \emptyset$, and $\mathrm{conv}(\emptyset) = \star$. Therefore, $\mathrm{conv}^A \colon (\mathcal{PD})^A \Rightarrow (C + 1)^A$ defined pointwise is natural as well. As a consequence [27], [54], we get a translation functor from NPLTS to convex NPLTS, and hence bisimilarity implies convex bisimilarity for NPLTS.

NPA. Nondeterministic Probabilistic automata, NPA, with observations in O are (for us in this paper) coalgebras for $F = O \times (C(\cdot + 1))^A$. We explain in Section 5.3 below how to move from (convex) NPLTS to NPA, which involves two steps: (1) Adding observations and (2) Dealing with termination.

We write $x \xrightarrow{a} m$ for $t(x)(a) = m$ with $a \in A, x \in X, m \in MX$ in a system or automaton with M-effects. For an LTS $t \colon X \to (\mathcal{P}X)^A$ we also write, as usual, $x \xrightarrow{a} y$ for $y \in t(x)(a)$ and $x \stackrel{a}{\not\to}$ if $t(x)(a) = \emptyset$; for an RPLTS $t \colon X \to (\mathcal{D}X + 1)^A$, we may also write $x \xrightarrow{a}_p y$ for $t(x)(a)(y) = p$ and again $x \stackrel{a}{\not\to}$ if $t(x)(a) = \star$. Note that in all our examples of systems and automata there is an implicit finite branching property ensured by the use of $\mathcal{P}$, $\mathcal{D}$ and C involving only finite subsets, finitely supported distributions, and finitely generated convex sets.

5.2. Generalised Determinisation

The construction of generalised determinisation was originally discovered in [30], [56]. It enables us to obtain trace semantics for coalgebras of type $c \colon X \to FMX$ where F is a functor and M a monad. The result is a determinised coalgebra $c^{\#} \colon MX \to FMX$ and the semantics is derived from behavioural equivalence for F-coalgebras.

Let $c \colon X \to FMX$ be a coalgebra and $\lambda \colon MF \Rightarrow FM$ a distributive law of the monad M over the functor F. Such a λ is a natural transformation that commutes appropriately with the unit and the multiplication of M, i.e., $\lambda \circ \eta = F\eta$ and $\lambda \circ \mu = F\lambda \circ \lambda \circ M\lambda$. Then the determinisation is the coalgebra

$$c^{\sharp} = F\mu \circ \lambda \circ Mc. \tag{2}$$

It is easy to show that $c^{\sharp} \circ \eta = c$ which justifies the notation $c^{\sharp}$: The carrier MX carries an M-algebra, the free one generated by X, FMX also does, $F\mu \circ \lambda$ is an M-algebra, and $c^{\sharp}$ is the unique extension of c to a homomorphism from the free M-algebra (MX, μ) to the M-algebra $(FMX, F\mu \circ \lambda)$.

We obtain behavioral equivalence on MX via the final coalgebra morphism $[\![\cdot]\!]_{c^{\sharp}}$ into the final coalgebra for F: for m, n in MX, $m \sim n$ iff $[\![m]\!]_{c^{\sharp}} = [\![n]\!]_{c^{\sharp}}$. This in turn induces an equivalence on X, via the unit of the monad η: for $x, y \in X$, $x \equiv y$ iff $\eta(x) \sim \eta(y)$. If F is such that a final F-coalgebra does not exist, we can still define $\equiv$ via behavioural equivalence by: for $x, y \in X$, $x \equiv y$ iff $\eta(x) \approx \eta(y)$. This induced semantics $\equiv$ on X is what we call the *trace semantics via determinisation*.

Determinising automata with M-effects and observations in O. In this paper, we only consider determinisation of automata with M-effects and observations in O. Hence, FM-coalgebras for the Moore-automata functor $F = O \times (\cdot)^A$, where O is a set of observations. The following proposition shows that determinising automata with M-effects and observations in O is always possible when the observations carry an M-algebra [30], [57].

Proposition 11. *For an Eilenberg-Moore algebra $a\colon MO \to O$, for $F = O\times(\cdot)^A$ and any monad M on* $\mathbf{Sets}$ *there is a canonical distributive law $\lambda_X : MF \Rightarrow FM$ given by*

$$M\big(O\times X^A\big) \xrightarrow{\langle M\pi_1, M\pi_2\rangle} MO\times M(X^A) \xrightarrow{a\times \mathsf{st}} O\times(MX)^A$$

where st is the map st: $M(X^A) \to (MX)^A$ defined, for all labels $a \in A$, by $\mathsf{st}(\varphi)(a) = M\mathsf{ev}_a(\varphi)$ with $\mathsf{ev}_a\colon X^A \to X$ the evaluation map defined as $\mathsf{ev}_a(\varphi) = \varphi(a)$. $\square$

As a consequence, we can determinise $c = \langle o, t\rangle\colon X \to O \times (MX)^A$ to $c^\sharp = \langle o^\sharp, t^\sharp\rangle$ where $o^\sharp = a \circ Mo$ and $t^\sharp = \mu_X^A \circ st \circ Mt$. The final coalgebra for the determinisation of automata with M-effects and observations in O is carried by the O-weighted languages over alphabet A, i.e., maps $A^* \to O$. Unfolding the inductive definition of the final coalgebra semantics for automata with M-effects and observations in O, see e.g. [28], gives $[\![\eta(x)]\!]_{c^\sharp}(\varepsilon) = o^\sharp(x)$ and $[\![\eta(x)]\!]_{c^\sharp}(aw) = [\![t^\sharp(x)(a)]\!]_{c^\sharp}(w)$.

Knowing that (Σ, E) is a presentation for the monad M, we can write the algebraic structure, and hence the determinisation concretely as follows. For an n-ary operation symbol $f \in \Sigma$ and a (Σ, E)-algebra $\mathbb{A} = (A, \Sigma_A)$ we write f_A for the n-ary operation on A that is the interpretation of f. We have

$$\begin{aligned}&f_{FMX}(\langle o_1, f_1\rangle, \dots, \langle o_n, f_n\rangle) = \\ &\quad\langle f_O(o_1, \dots, o_n), (a \mapsto f_{MX}(f_1(a), \dots, f_n(a)))\rangle.\end{aligned}$$

Therefore, for a coalgebra $c\colon X \to FMX$, we have that $c^\sharp = \langle o^\sharp, t^\sharp\rangle$ is inductively defined on the structure of the Σ-terms by $o^\sharp(x) = o(x)$, $t^\sharp(x) = t(x)$ and

$$\begin{aligned}o^\sharp(f_{MX}(t_1, \dots, t_n)) &= f_O(o^\sharp(t_1), \dots, o^\sharp(t_n)) \qquad (3)\\ t^\sharp(f_{MX}(t_1, \dots, t_n))(a) &= f_{MX}(t^\sharp(t_1)(a), \dots, t^\sharp(t_n)(a))\end{aligned}$$

Example 12. *Applying this construction to $F = 2 \times (\cdot)^A$ and $M = \mathcal{P}$, one transforms $c\colon X \to 2 \times (\mathcal{P}X)^A$ into $c^\sharp\colon \mathcal{P}X \to 2 \times (\mathcal{P}X)^A$. The former is a nondeterministic automaton and the latter is a deterministic automaton which has $\mathcal{P}X$ as states space. In [30], see also [57], it is shown that, using the distributive law from Proposition 11, as $2 = \mathcal{P}1$ is the carrier of the free $\mathcal{P}$-algebra, this amounts exactly to the standard determinisation from automata theory and justifies the term* generalised determinisation*. The obtained semantics is language equivalence.*

It is worth to mention that both the determinised coalgebra $c^\sharp\colon MX{\to}FMX$ and the final F-coalgebra are actually *bialgebras* [58], [59], roughly they are both an M-algebra and an F-coalgebra. Moreover, the unique coalgebra morphism $[\![\cdot]\!]_{c^\sharp}\colon MX \to O^{A^*}$ is also an M-algebra homomorphism. The latter entails the first item of the following.

Theorem 13 ([30], [60])**.** *The following properties hold for any coalgebra $c\colon X{\to}FMX$ and its determinisation $c^\sharp\colon MX{\to}FMX$:*

1) *Behavioural equivalence for $(MX, c^\sharp)$ is a congruence w.r.t. the algebraic structure of M.*
2) *Behavioural equivalence for (X, c) implies trace semantics via determinisation.*
3) *Up-to context is a compatible [38] proof technique.*

The second item will be used later in Section 6 to show that convex bisimilarity implies trace equivalence for NPLTS.

5.3. From Systems to Automata

Dealing with automata, i.e., having observations, is crucial for determinisation. Starting from an LTS $t\colon X \to (\mathcal{P}X)^A$, we can add observations in $2 = \mathcal{P}1$ in the simplest possible way, making every state an accepting state:

$$o = (X \xrightarrow{!} 1 \xrightarrow{\eta_1} \mathcal{P}1 = 2)$$

and determinise the NA $\langle o, t\rangle\colon X \to 2 \times (\mathcal{P}X)^A$. The induced semantics $\equiv^{LTS}$ on the state space X is the standard trace semantics for LTS [61].

This same approach can be applied in the case of any system with M-effects $t\colon X \to (MX)^A$. We can add observations in $O = M1$ by

$$o = (X \xrightarrow{!} 1 \xrightarrow{\eta_1} M1),$$

determinise the automaton $\langle o, t\rangle$ with M-effects using the free algebra on $M1$, and obtain the trace semantics after determinisation $\equiv$.

From NPLTS to NPA. In order to define trace semantics for NPLTS via generalised determinisation, we need to transform them into NPA which are automata with $C(\cdot{+}1)$-effects. We proceed in two steps: we transform an NPLTS to a system with $C(\cdot{+}1)$-effects, and then add observations via the general recipe of this section. Given an NPLTS $t\colon X \to (\mathcal{P}\mathcal{D}X)^A$ we first transform it into the convex NPLTS $X \xrightarrow{t} (\mathcal{P}\mathcal{D}X)^A \xrightarrow{\mathrm{conv}^A} (CX{+}1)^A$ and then employ the distributive law ι from Section 2.3 to obtain

$$\begin{aligned}\bar{t} = \big(X &\xrightarrow{t} (\mathcal{P}\mathcal{D}X+1)^A \xrightarrow{\mathrm{conv}^A} (CX+1)^A \qquad (4)\\ &\xrightarrow{\iota^A} (C(X+1))^A\big)\end{aligned}$$

Note that $\bar{t}$ is a system with $C(\cdot+1)$-effects. Moreover, by construction, NPLTS-bisimilarity for t implies convex bisimilarity, and further convex bisimilarity implies behavioural equivalence for the resulting system with $C(\cdot{+}1)$-effects $\bar{t}$. Finally, we add observations as prescribed above:

$$\bar{o} = \big(X \xrightarrow{!} 1 \xrightarrow{\eta_1} C(1+1)\big) \qquad (5)$$

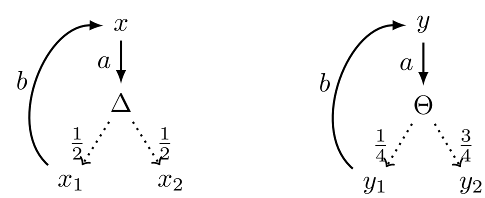
and get the desired automaton with $C(\cdot+1)$-effects and observations in $C(1+1)$. Adding such observations again preserves behavioural equivalence.

Inside vs outside termination. We have seen that, when moving from NPLTS to NPA, in particular when moving from convex NPLTS to NPA, we are not just adding an observation. We are also moving, via the ι distributive law, from the functor $C+1$ to the functor $C(\cdot+1)$. The reason why we do this can already be understood in the simpler case of RPLTS, where the monad $\mathcal{D}$ is used instead of C. We have that $\mathcal{D}+1$ is already a monad, and there is a monad map in both directions between $\mathcal{D}(\cdot+1)$ and $\mathcal{D}+1$. So we could take a $\mathcal{D}+1$-algebra and perform a determinisation with respect to $\mathcal{D}+1$. There is however an undesired consequence of doing so, as illustrated by the following example.

Example 14. *Trace semantics for RPLTS is defined in a similar way, see the construction in [62], [63]. An RPLTS $t_*\colon X \to (\mathcal{D}X+1)^A$ can similarly be transformed to a system with $\mathcal{D}(\cdot+1)$-effects using the distributive law ι:*

$$t = X \xrightarrow{t_*} (\mathcal{D}X+1)^A \xrightarrow{\iota^A} (\mathcal{D}(X+1))^A.$$

Consider the following RPLTS.



The states x and y should not be trace equivalent, since x has probability $\frac{1}{2}$ of performing trace ab, and y has probability $\frac{1}{4}$ of performing trace ab. Let us look at what happens, however, if we determinise this system (seen as the $(\mathcal{D}+1)^A$ coalgebra t_*) with respect to the monad $\mathcal{D}+1$. The determinised transition function $t_*^\sharp$ will give us states in $\mathcal{D}X+1$, i.e., states that are either full distributions or the element $\star \in 1$ and we have

$$t_*^\sharp(x)(a) = x_1 +_{\frac{1}{2}} x_2 \qquad t_*^\sharp(y)(a) = y_1 +_{\frac{1}{4}} y_2$$

However,
$$t_*^\sharp(x_1 +_{\frac{1}{2}} x_2)(b) = t_*(x_1)(b) +_{\frac{1}{2}} t_*(x_2)(b) = \star$$
$$t_*^\sharp(y_1 +_{\frac{1}{4}} y_2)(b) = t_*(y_1)(b) +_{\frac{1}{4}} t_*(y_2)(b) = \star$$

Hence, whatever $(\mathcal{D}+1)$-algebra of observation we take, these states in the determinised system will return the same observation, i.e., $o^\sharp(x)(ab) = o^\sharp(y)(ab)$. As a consequence, x and y will be equivalent.

Hence, moving to a monad with termination inside is a fundamental step in our construction, if we want to distinguish processes such as those in the previous example.

However, there are cases in which determinising with respect to two different monads and algebras leads to the same semantics, as shown in the next example.

Example 15. *As described above, we turn an RPLTS into an automaton with $\mathcal{D}(\cdot+1)$-effects with observations in $[0,1] = \mathcal{D}(1+1)$ equipped with the the free algebra generated by 1. The observation function $o\colon X \to [0,1]$ maps every state $x \in X$ into the element $1 \in [0,1]$. The function $[\![\cdot]\!]_{c^\sharp} \circ \eta\colon X \to [0,1]^{A^*}$ obtained via the generalised determinisation of $c = \langle o,t\rangle$ assigns to each state $x \in X$ and trace $w \in A^*$ the probability of reaching from x any other state via w. We write $\equiv^{RP}$ for the induced trace equivalence.*

Interestingly, (Rabin) probabilistic automata *[31] are defined slightly differently: these are automata with $\mathcal{D}$-effects and observations in $[0,1]$, $\langle o,t\rangle\colon X \to [0,1]\times(\mathcal{D}X)^A$ (see [30]). The set of observations is the same, but transitions go in distributions rather than in subdistributions. The theorem of the next section guarantees that* only *the algebra of observations matters for the resulting semantics, so using $\mathcal{D}$ in place of $\mathcal{D}(\cdot+1)$ does not change the obtained equivalence which in both cases coincides with the probabilistic language equivalence of [31].*

5.4. Invariance of the Semantics

We next state a theorem that guarantees invariance of the trace semantics via determinisation for automata with M-effects and observations in O, under controlled changes of the monad or the algebra of observations. The proofs of the invariance theorem and its corollary are in [41] Appendix F.

Theorem 16 (Invariance Theorem)**.** *Let (M,η,μ) be a monad and $a\colon MO \to O$ an M-algebra. Let $c = \langle o,t\rangle\colon X \to O\times(MX)^A$ be an automaton with M-effects and observations in O and $[\![\cdot]\!]\colon MX \to O^{A^*}$ the semantic map induced by the generalised determinisation wrt. a, i.e., $[\![\cdot]\!] = [\![\cdot]\!]_{c^\sharp}$*

1) **Transitions:** *Let $(\hat{M},\hat{\eta},\hat{\mu})$ be a monad and $\sigma\colon M \Rightarrow \hat{M}$ a monad map. Let $\hat{a}\colon \hat{M}O \to O$ be an $\hat{M}$-algebra. Consider the coalgebra*
$$\hat{c} = \langle o,\hat{t}\rangle = \langle o, \sigma_X^A \circ t\rangle\colon X \to O\times(\hat{M}X)^A$$
and let $\hat{[\![\cdot]\!]}\colon \hat{M}X \to O^{A^*}$ be the semantic map induced by its generalised determinisation wrt. $\hat{a}$. If $a = \hat{a}\circ\sigma_O$, then $[\![\cdot]\!]\circ\eta_X = \hat{[\![\cdot]\!]}\circ\hat{\eta}_X$.

2) **Observations:** *Let $\hat{a}\colon M\hat{O} \to \hat{O}$ be an M-algebra and let $h\colon (O,a) \to (\hat{O},\hat{a})$ be an M-algebra morphism. Consider the coalgebra*
$$\hat{c} = \langle\hat{o},t\rangle = \langle h\circ o,t\rangle\colon X \to \hat{O}\times(MX)^A$$
and let $\hat{[\![\cdot]\!]}\colon TX \to \hat{O}^{A^*}$ be induced by the generalised determinisation wrt. $\hat{a}$. Then $\hat{[\![\cdot]\!]} = h^{A^*}\circ[\![\cdot]\!]$. □

Corollary 17. *Let (M,η,μ) be a submonad of $(\hat{M},\hat{\eta},\hat{\mu})$ via an injective monad map $\sigma\colon M \Rightarrow \hat{M}$. Let $t\colon X \to (MX)^A$ be a system with M-effects and let $\hat{t}$ be the system with $\hat{M}$-effects $\sigma_X^A\circ t\colon X \to (\hat{M}X)^A$. Let $o = \big(X \xrightarrow{!} 1 \xrightarrow{\eta_1} M1\big)$ and $\hat{o} = \big(X \xrightarrow{!} 1 \xrightarrow{\hat{\eta}_1} \hat{M}1\big)$ and $\equiv, \hat{\equiv} \subseteq X\times X$ be the corresponding trace equivalences after determinisation of $\langle o,t\rangle$, $\langle\hat{o},\hat{t}\rangle$, respectively. Then $\equiv = \hat{\equiv}$.* □

6. May / Must Traces for NPLTS

In this section, we put all the pieces together and give the definitions of may, must, and may-must trace semantics for NPLTS using generalised determinisation. We work with the monad $T_{\mathcal{PCS}} = C(\cdot + 1)$ and consider its two quotients $T_{\mathcal{CSB}}$ and $T_{\mathcal{CST}}$. Each of these choices gives us a trace equivalence via determinisation. We start with the notion of may-must traces.

May-must trace equivalence. Given an NPLTS $t\colon X \to (\mathcal{PD}X)^A$, let $(\bar{o}, \bar{t}\,)$ be the automaton with $T_{\mathcal{PCS}}$-effects and observations in $T_{\mathcal{PCS}}1$ as in Equation (4) and Equation (5). Let $\langle \bar{o}^\sharp, \bar{t}^\sharp \rangle$ be the determinisation of $\langle o, t\rangle$ using the free $T_{\mathcal{PCS}}$-algebra, i.e., by Proposition 8, the min-max interval pointed convex semilattice $\mathbb{M}_{\mathcal{I},[0,0]}$, on $T_{\mathcal{PCS}}1$. We write $[\![\cdot]\!]$ for the semantics map from $T_{\mathcal{PCS}}X \to (T_{\mathcal{PCS}}1)^{A^*}$ and $\equiv$ for the corresponding trace equivalence on X. We call this equivalence *may-must trace equivalence* for the original NPLTS.

Using the presentation of the monad, as in Equation (3), recalling that $\bar{o}(x) = [1,1]$ we can spell out the inductive definition of the determinisation:

$$\bar{o}^\sharp(S) \quad = \quad \begin{cases} [1,1] & \text{if } S = x; \\ [0,0] & \text{if } S = \star; \\ S_1 \,\text{min-max}\, S_2 & \text{if } S = S_1 \oplus S_2; \\ S_1 +_p S_2 & \text{if } S = S_1 +_p S_2. \end{cases}$$

$$\bar{t}^\sharp(S)(a) \quad = \quad \begin{cases} \bar{t}(x)(a) & \text{if } S = x; \\ \star & \text{if } S = \star; \\ \bar{t}^\sharp(S_1)(a) \oplus \bar{t}^\sharp(S_2)(a) & \text{if } S = S_1 \oplus S_2; \\ \bar{t}^\sharp(S_1)(a) +_p \bar{t}^\sharp(S_2)(a) & \text{if } S = S_1 +_p S_2. \end{cases}$$

May trace equivalence and must trace equivalence. Now one may want to treat termination in a different way and exploit the monads $T_{\mathcal{CSB}}$ and $T_{\mathcal{CST}}$ discussed in Section 4. Given the monad morphisms $q_B\colon T_{\mathcal{PCS}} \Rightarrow T_{\mathcal{CSB}}$ and $q_T\colon T_{\mathcal{PCS}} \Rightarrow T_{\mathcal{CST}}$ quotienting $T_{\mathcal{PCS}}$ by (B) and (T), respectively, one can construct the transition functions

$$\bar{t}_B = q_B^A \circ \bar{t}\colon X \to (T_{\mathcal{CSB}}X)^A$$
$$\bar{t}_T = q_T^A \circ \bar{t}\colon X \to (T_{\mathcal{CST}}X)^A.$$

For the observations, we always use the general recipe of Section 5.3 and take the observation functions:

$$\bar{o}_B = \big(X \xrightarrow{!} 1 \xrightarrow{\eta_1} T_{\mathcal{CSB}}1\big) \quad \bar{o}_T = \big(X \xrightarrow{!} 1 \xrightarrow{\eta_1} T_{\mathcal{CST}}1\big).$$

Recall from Proposition 9 and Proposition 10 that $\mathbb{M}\text{ax}_B = ([0,1], \max, +_p, 0)$ and $\mathbb{M}\text{in}_T = ([0,1], \min, +_p, 0)$ are, the free convex semilattice with bottom and, respectively, with top, generated by the singleton set 1. Therefore these algebraic structures will be used for the determinisations. We have, $\bar{o}_B^\sharp\colon T_{\mathcal{CSB}}X \to [0,1]$ and $\bar{o}_T^\sharp\colon T_{\mathcal{CST}}X \to [0,1]$ are given as follows, since $\bar{o}_B(x) = 1$ and $\bar{o}_T(x) = 1$:

$$\bar{o}_B^\sharp(S) \quad = \quad \begin{cases} 1 & \text{if } S = x; \\ 0 & \text{if } S = \star; \\ S_1 \max S_2 & \text{if } S = S_1 \oplus S_2; \\ S_1 +_p S_2 & \text{if } S = S_1 +_p S_2. \end{cases}$$

$$\bar{o}_T^\sharp(S) \quad = \quad \begin{cases} 1 & \text{if } S = x; \\ 0 & \text{if } S = \star; \\ S_1 \min S_2 & \text{if } S = S_1 \oplus S_2; \\ S_1 +_p S_2 & \text{if } S = S_1 +_p S_2. \end{cases}$$

The determinisation of the transition function $\bar{t}_B^\sharp\colon T_{\mathcal{CSB}}X \to (T_{\mathcal{CSB}}X)^A$ and $\bar{t}_T^\sharp\colon T_{\mathcal{CST}}X \to (T_{\mathcal{CST}}X)^A$ are defined in the same way like $\bar{t}^\sharp$ above.

The coalgebras $\langle \bar{o}_B^\sharp, \bar{t}_B^\sharp \rangle$ and $\langle \bar{o}_T^\sharp, \bar{t}_T^\sharp \rangle$ give rise to morphisms $[\![\cdot]\!]_B\colon T_{\mathcal{CSB}}X \to [0,1]^{A^*}$ and $[\![\cdot]\!]_T\colon T_{\mathcal{CST}}X \to [0,1]^{A^*}$ and corresponding behavioural equivalences: $\equiv_B$ and $\equiv_T$. We call $\equiv_B$ the *may trace equivalence* for the NPLTS, and $\equiv_T$ the *must trace equivalence*.

Example 18. *Consider the convex closure of the NPLTS from Figure 1. We can syntactically describe the sets of subdistributions reached by a state when performing a transition as follows:*

$$x \xrightarrow{a} x_1 \oplus (x_3 +_{\frac{1}{2}} x_2)$$
$$y \xrightarrow{a} y_1 \oplus (y_4 +_{\frac{1}{2}} y_2) \oplus ((y_2 +_{\frac{1}{2}} y_4) +_{\frac{1}{2}} y_3)$$
$$x_1 \xrightarrow{b} x +_{\frac{1}{2}} x_3 \qquad y_1 \xrightarrow{b} y +_{\frac{1}{2}} y_4$$
$$x_2 \xrightarrow{b} x_3 \qquad x_2 \xrightarrow{c} x \qquad y_2 \xrightarrow{b} y_4 \qquad y_3 \xrightarrow{c} y$$

In the determinised system, we have

$$x \xrightarrow{a} S_1 \xrightarrow{b} S_2 \qquad y \xrightarrow{a} S_1' \xrightarrow{b} S_2'$$

for $S_1 = x_1 \oplus (x_3 +_{\frac{1}{2}} x_2) \qquad S_2 = (x +_{\frac{1}{2}} x_3) \oplus (\star +_{\frac{1}{2}} x_3)$

$$S_1' = y_1 \oplus (y_4 +_{\frac{1}{2}} y_2) \oplus ((y_2 +_{\frac{1}{2}} y_4) +_{\frac{1}{2}} y_3)$$
$$S_2' = (y +_{\frac{1}{2}} y_4) \oplus (\star +_{\frac{1}{2}} y_4) \oplus ((y_4 +_{\frac{1}{2}} \star) +_{\frac{1}{2}} \star)$$

Consider now the observations associated to the terms in the may-must semantics. We have $\bar{o}^\sharp(x) = [1,1] = \bar{o}^\sharp(y)$ *and hence*

$$\bar{o}^\sharp(S_1) = [1,1]\,\text{min-max}([1,1] +_{\frac{1}{2}} [1,1]) = [1,1].$$

Analogously, $\bar{o}^\sharp(S_1') = [1,1]$. *Furtheron*

$$\bar{o}^\sharp(S_2) = ([1,1] +_{\frac{1}{2}} [1,1])\,\text{min-max}([0,0] +_{\frac{1}{2}} [1,1]) = [\frac{1}{2}, 1]$$

and in the same way we derive $\bar{o}^\sharp(S_2') = [\frac{1}{4}, 1]$.

Hence, x and y are not may-must trace equivalent: $[\![x]\!](ab) = \bar{o}^\sharp(S_2) \neq \bar{o}^\sharp(S_2') = [\![y]\!](ab)$.

However, using $\mathbb{M}\text{ax}_B$, we get $\bar{o}_B^\sharp(S_2) = \bar{o}_B^\sharp(S_2')$ as the intervals obtained via the may-must observation over S_2, S_2' have the same upper bound 1, which is the value returned by both $\bar{o}_B^\sharp(S_2)$ and $\bar{o}_B^\sharp(S_2')$. Hence, $[\![x]\!]_B(ab) = \bar{o}_B^\sharp(S_2) =$

$\bar{o}^{\sharp}_B(S'_2) = [\![y]\!]_B(ab)$. *More generally, it holds that x and y are may trace equivalent. We can elegantly prove this by using up-to techniques, as shown in [41] Appendix A.*

The following properties follow automatically from our abstract construction: see Theorem 13 and the discussions in Section 5.3.

Theorem 19. *The following properties hold for NPLTS:*

1) *Each of the three trace equivalences is a congruence w.r.t. $+_p$, $\oplus$ and $\star$.*
2) *Both bisimilarity and convex bisimilarity imply each of the three trace equivalences.*
3) *Up-to context is compatible (see [41] Appendix A) for each of the three equivalences.* $\square$

We might have performed the generalised determinisation in a number of different ways, for instance by eliminating conv from the definition of $\bar{t}$. In [41] Appendix G, we show that Theorem 16 guarantees that many different construction always lead to our semantics. In the same appendix we give also a simple concrete description of the final-coalgebra bialgebra of probabilistic traces.

Backward compatibility. We now state the backward compatibility of our semantics with the corresponding trace semantics for LTS and RPLTS. The proof follows from Corollary 17, since: (1) $\mathcal{P} \cong T_{\mathcal{SB}}$ for the theory $\mathcal{SB}$ of semilattices with bottom and we show that there is an injective monad map $T_{\mathcal{SB}} \Rightarrow T_{\mathcal{CSB}}$; and (2) The natural transformation $\mathrm{conv} \circ \eta^{\mathcal{P}_{ne}} : \mathcal{D} \Rightarrow C$ is an injective monad map and hence, by Lemma 1, there is an injective monad map $\mathcal{D}(\cdot + 1) \Rightarrow C(\cdot + 1)$.

Theorem 20. *Trace semantics $\equiv^{LTS}$ for LTS coincides with may trace semantics after determinisation $\equiv_B$ of the LTS seen as NPLTS. Trace semantics $\equiv^{RP}$ for RPLTS coincides with each of the three (may, must, and may-must) trace semantics $\equiv$ of the RPLTS seen as NPLTS.* $\square$

For LTS, one can also study the variants corresponding to must and may-must trace semantics, that have not been studied in the literature. We define them in [41] Appendix G, and show backward compatibility results for them as well.

7. From the global to the local perspective

Usually trace semantics for NPLTS is defined in terms of *schedulers*, or *resolutions*: intuitively, a scheduler resolves the nondeterminism by choosing, at each step of the execution of an NPLTS, one of its possible transitions; the transition systems resulting from these choices are called resolutions.

This perspective on trace semantics is somehow opposed to ours, where the generalised determinisation keeps track of all possible executions at once. In this sense, the determinisation provides a perspective which is *global*, opposite to those of resolutions that are *local*. In this section, we show that our semantics can be characterised through such local views, by means of resolutions, defined as follows.

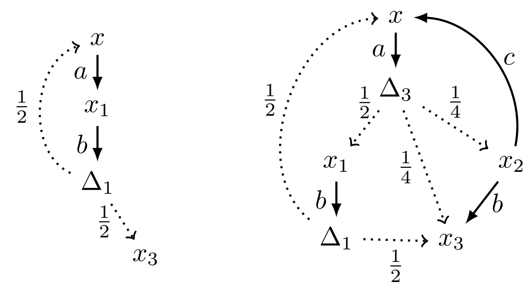


Figure 2. The resolutions $\mathcal{R}_1$ (left) and $\mathcal{R}_2$ (right)

Definition 21. *Let $t\colon X \to (\mathcal{PD}X)^A$ be an NPLTS. A* (randomized) resolution *for t is a triple $\mathcal{R} = (Y, \mathrm{corr}, r)$ where Y is a set of states,* $\mathrm{corr}\colon Y \to X$ *is* the correspondence function, *and $r\colon Y \to (\mathcal{D}Y + 1)^A$ is an RPLTS such that for all $y \in Y$ and $a \in A$,*

1) *$r(y)(a) = \star$ iff $t(\mathrm{corr}(y))(a) = \star$,*
2) *if $r(y)(a) \neq \star$ then $\mathcal{D}(\mathrm{corr})(r(y)(a)) \in \mathrm{conv}(t(\mathrm{corr}(y))(a))$.*

Intuitively, this means that a resolution of an NPLTS is built from the original system by discarding internal nondeterminism (the possibility to perform multiple transitions labelled with the same action) and in such a way that the structure of the original system is preserved.

Example 22. *Consider the NPLTS on the left of Figure 1. Figure 2 illustrates two resolutions for it, both having the identity as correspondence function. In the resolution $\mathcal{R}_1$, the nondeterministic choice of x is resolved by choosing the leftmost a-transition. Instead, the resolution $\mathcal{R}_2$ is obtained by taking a convex combination of the two distributions δ_{x_1} and Δ_1, assigning one half probability to each of them.*

The reason why we take arbitrary corr functions, rather than just injective ones, is that the original NPLTS might contain cycles, in which case we want to allow the resolution to take different choices at different times (see [41] Appendix H).

Given a resolution $\mathcal{R} = (Y, \mathrm{corr}, r)$, we define the function $\mathrm{prob}_{\mathcal{R}}\colon Y \to [0,1]^{A^*}$ inductively for all $y \in Y$ and all $w \in A^*$ as

$$\mathrm{prob}_{\mathcal{R}}(y)(\varepsilon) = 1;$$
$$\mathrm{prob}_{\mathcal{R}}(y)(aw) =$$
$$\begin{cases} 0 & \text{if } r(y)(a) = \star; \\ \sum_{y' \in \mathrm{supp}(\Delta)} \Delta(y') \cdot \mathrm{prob}_{\mathcal{R}}(y')(w) & \text{if } r(y)(a) = \Delta. \end{cases}$$

Intuitively, for all states $y \in Y$, $\mathrm{prob}_{\mathcal{R}}(y)(w)$ gives the probability of y performing the trace w. For instance, in the resolutions in Figure 2, $\mathrm{prob}_{\mathcal{R}_1}(x)(abab) = \frac{1}{2}$ and $\mathrm{prob}_{\mathcal{R}_2}(x)(abab) = \frac{3}{16}$.

Now, given an NPLTS (X, t), define $\lfloor\!\lfloor\cdot\rfloor\!\rfloor\colon X \to [0,1]^{A^*}$ with, for all $x \in X$ and $w \in A^*$, $\lfloor\!\lfloor x\rfloor\!\rfloor(w)$ equal to

$$\bigsqcup\{\mathrm{prob}_{\mathcal{R}}(y)(w) \mid \mathcal{R} = (Y, \mathrm{corr}, r) \text{ is a resolution of } (X,t) \text{ and } \mathrm{corr}(y) = x\}.$$

Similarly, we define $\lceil\!\lceil x \rceil\!\rceil(w)$ as

$$\bigsqcap\{\mathrm{prob}_{\mathcal{R}}(y)(w) \mid \mathcal{R} = (Y, \mathrm{corr}, r) \text{ is a resolution of } (X,t) \text{ and } \mathrm{corr}(y) = x\}.$$

The following theorem states that the global view of trace semantics developed in Section 6 coincides with the trace semantics defined locally via resolutions.

Theorem 23 (Global/local correspondence)**.** *Let (X,t) be an NPLTS. For all $x \in X$ and $w \in A^*$, it holds that*

$$[\![x]\!](w) = [\, \lceil\!\lceil x \rceil\!\rceil(w), \lfloor\!\lfloor x \rfloor\!\rfloor(w)\,].$$

Corollary 24. *Let (X,t) be an NPLTS. For all $x \in X$ and $w \in A^*$, $[\![x]\!]_B(w) = \lfloor\!\lfloor x \rfloor\!\rfloor(w)$ and $[\![x]\!]_T(w) = \lceil\!\lceil x \rceil\!\rceil(w)$.*

Theorem 23 and Corollary 24 provide a characterisation of $\equiv$, $\equiv_B$ and $\equiv_T$ in terms of resolutions. We next show that $\equiv_B$ coincides with the randomized $\sqcup$-trace equivalence investigated in [40] and inspired by [25], [26].

Coincidence with randomized $\sqcup$-trace equivalence. Let $t\colon X \to (\mathcal{P}\mathcal{D}X)^A$ be an NPLTS. A *fully probabilistic resolution* for t is a triple $\mathcal{R} = (Y, \mathrm{corr}, r)$ such that Y is a set, $\mathrm{corr}\colon Y \to X$, and $r\colon Y \to (A \times \mathcal{D}Y) + 1$ such that for every $y \in Y$ and $a \in A$ it holds:
if $r(y) = \langle a, \Delta\rangle$ then $\mathcal{D}(\mathrm{corr})(\Delta) \in \mathrm{conv}(t(\mathrm{corr}(y))(a))$.

While resolutions resolve only internal nondeterminism, fully probabilistic resolutions resolve both internal and external nondeterminism. Indeed, in a resolution a state can perform transitions with different labels, while in a fully probabilistic resolution a state can perform at most one transition. Moreover, a state y in a fully probabilistic resolutions might not perform any transition (i.e., $r(y) = \star$), even if the corresponding state $\mathrm{corr}(y)$ may perform a transition (i.e., $t(\mathrm{corr}(y))(a) \neq \star$ for some a).

Example 25. *As in Example 22, consider the NPLTS on the left of Figure 1. The resolution $\mathcal{R}_1$ in Figure 2 is a fully probabilistic resolution, while $\mathcal{R}_2$ is not, since x_2 is allowed to perform more than one transition, even if labelled by different actions. Other examples of fully probabilistic resolutions are given in [41] Appendix B.2.*

As for resolutions, we can define $\mathrm{prob}_{\mathcal{R}}\colon Y \to [0,1]^{A^*}$ for $\mathcal{R} = (Y, \mathrm{corr}, r)$ a fully probabilistic resolution inductively for all $y \in Y$ and all $w \in A^*$ as

$$\mathrm{prob}_{\mathcal{R}}(y)(\varepsilon) = 1;$$
$$\mathrm{prob}_{\mathcal{R}}(y)(aw) = \begin{cases} \sum_{y' \in \mathrm{supp}(\Delta)} \Delta(y') \cdot \mathrm{prob}_{\mathcal{R}}(y')(w) & \text{if } r(y) = \langle a, \Delta\rangle, \\ 0 & \text{otherwise.} \end{cases}$$

Given an NPLTS (X,t), we define for all $x \in X$ and $w \in A^*$ $\lfloor\!\lfloor x \rfloor\!\rfloor_{fp}(w)$ as

$$\bigsqcup\{\mathrm{prob}_{\mathcal{R}}(y)(w) \mid \mathcal{R} = (Y, \mathrm{corr}, r) \text{ is a fully probabilistic resolution of } (X,t) \text{ and } \mathrm{corr}(y) = x\}.$$

In [40] (following [25], [26]), two states x and y are defined to be *randomized $\sqcup$-trace equivalent* whenever $\lfloor\!\lfloor x \rfloor\!\rfloor_{fp}(w) = \lfloor\!\lfloor y \rfloor\!\rfloor_{fp}(w)$, $w \in A^*$.[2] The following proposition guarantees that such equivalence coincides with $\equiv_B$.

Proposition 26. *Let (X,t) be an NPLTS. For all $x \in X$ and $w \in A^*$, it holds that $[\![x]\!]_B(w) = \lfloor\!\lfloor x \rfloor\!\rfloor(w) = \lfloor\!\lfloor x \rfloor\!\rfloor_{fp}(w)$.*

Remark 27. *The correspondence in Proposition 26 does not hold when infima are considered, instead of suprema. Indeed define $\lceil\!\lceil x \rceil\!\rceil_{fp}(w)$ as expected, namely, by replacing $\bigsqcup$ with $\bigsqcap$ in $\lfloor\!\lfloor x \rfloor\!\rfloor_{fp}(w)$. Then for any state x of an arbitrary NPLTS it holds that $\lceil\!\lceil x \rceil\!\rceil_{fp}(w) = 0$ for all $w \neq \varepsilon$. To see this, observe that $\mathcal{R}' = (\{y\}, \mathrm{corr}', r')$ with $\mathrm{corr}'(y) = x$ and $r'(y) = \star$ is always a fully probabilistic resolution, and that $\mathrm{prob}_{\mathcal{R}'}(y)(w) = 0$.*

To avoid this problem, one typically modifies the definition of $\lceil\!\lceil \cdot \rceil\!\rceil_{fp}$ by restricting only to those fully probabilistic resolutions that can perform a certain trace (see e.g. [25], [26]). Instead, with our notion of resolution based on RPLTSs (Definition 21), this problems does not arise and the definition of $\lceil\!\lceil \cdot \rceil\!\rceil$ is totally analogous to the one of $\lfloor\!\lfloor \cdot \rfloor\!\rfloor$.

Why may, must, may-must? Trace equivalences as testing equivalences. The notion of resolution is at the basis not just of the definitions of trace equivalences for NPLTS investigated in the literature, but also of testing equivalences for nondeterministic and probabilistic processes [33], [34], [35], [36]. In testing equivalences, we say that x, y are may testing equivalent if, for every test, they have the same greatest probabilities of passing the test, with respect to any resolution $\mathcal{R}$ of the system resulting from the interaction between the test and the NPLTS. Analogously, x, y are must testing equivalent if the smallest probabilities coincide, and the may-must testing equivalence requires both the greatest and the smallest probabilities to coincide.

Now, take tests to be finite traces, and the probability of passing a given test in a resolution as the probability of performing the trace in the resolution. Then it becomes clear, by the correspondence between the local and the global view proven in Theorem 23, that each of our three trace equivalences indeed coincides with the corresponding testing equivalence, when tests are finite traces.

8. Conclusion

We developed an algebra-and-coalgebra-based trace theory for systems with nondeterminism and probability, that covers intricate trace semantics from the literature. The abstract approach sheds light on all choices and leaves no space for ad-hoc solutions.

Although the combination of nondeterminism and probability has been posing challenges to abstract approaches, this new algebraic theory of traces for NPLTS shows that it can be dealt with in a smooth and uniform way.

2. Actually, [25], [26], [40] use a notion of resolution which is equal to our fully-probabilistic resolution modulo a tiny modification due to a mistake in [25], [26], as confirmed by the authors in a personal communication.

References

[1] C. Baier and J. Katoen, *Principles of model checking*. MIT Press, 2008.

[2] H. Hermanns, J. Krcál, and J. Kretínský, "Probabilistic bisimulation: Naturally on distributions," in *Proc. CONCUR'14*, ser. LNCS, vol. 8704, 2014, pp. 249–265.

[3] M. Z. Kwiatkowska, G. Norman, and D. Parker, "Prism: Probabilistic symbolic model checker," in *Computer Performance Evaluation / TOOLS*. LNCS 2324, 2002, pp. 200–204.

[4] C. Dehnert, S. Junges, J. Katoen, and M. Volk, "A storm is coming: A modern probabilistic model checker," in *Proc. CAV 2017*, ser. LNCS, vol. 10427, 2017, pp. 592–600.

[5] M. Y. Vardi, "Automatic verification of probabilistic concurrent finite state programs," in *Foundations of Computer Science, 1985., 26th Annual Symposium on*. IEEE, 1985, pp. 327–338.

[6] H. A. Hansson, "Time and probability in formal design of distributed systems," *PhD thesis, Uppsala University*, 1991.

[7] R. Segala and N. Lynch, "Probabilistic simulations for probabilistic processes," *Nordic Journal of Computing*, vol. 2, no. 2, pp. 250–273, 1995.

[8] P. S. Castro, P. Panangaden, and D. Precup, "Equivalence relations in fully and partially observable markov decision processes," in *IJCAI*, 2009, pp. 1653–1658.

[9] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and Acting in Partially Observable Stochastic Domains." *Artif. Intell.*, 1998.

[10] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2009.

[11] C. Heunen, O. Kammar, S. Staton, and H. Yang, "A convenient category for higher-order probability theory," *CoRR*, vol. abs/1701.02547, 2017. [Online]. Available: http://arxiv.org/abs/1701.02547

[12] S. Staton, H. Yang, F. Wood, C. Heunen, and O. Kammar, "Semantics for probabilistic programming: higher-order functions, continuous distributions, and soft constraints," in *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, 2016, pp. 525–534. [Online]. Available: http://doi.acm.org/10.1145/2933575.2935313

[13] H. Hermanns, A. Parma, R. Segala, B. Wachter, and L. Zhang, "Probabilistic logical characterization," *Information and Computation*, vol. 209, no. 2, pp. 154–172, 2011.

[14] D. Varacca and G. Winskel, "Distributing probabililty over nondeterminism," *MSCS*, vol. 16, no. 1, pp. 87–113, 2006.

[15] K. Keimel and G. D. Plotkin, "Mixed powerdomains for probability and nondeterminism," *Logical Methods in Computer Science*, vol. 13, no. 1, 2017. [Online]. Available: https://doi.org/10.23638/LMCS-13(1:2)2017

[16] M. Mio, "Upper-expectation bisimilarity and łukasiewicz μ-calculus," in *Proc. FOSSACS'14*, ser. LNCS, vol. 8412, 2014, pp. 335–350.

[17] B. Jacobs, "Coalgebraic trace semantics for combined possibilitistic and probabilistic systems," *Electr. Notes Theor. Comput. Sci.*, vol. 203, no. 5, pp. 131–152, 2008.

[18] D. Varacca, *Probability, Nondeterminism and Concurrency. Two Denotational Models for Probabilistic Computation*. PhD Thesis, 2003.

[19] M. W. Mislove, "Nondeterminism and probabilistic choice: Obeying the laws," in *CONCUR 2000*. LNCS 1877, 2000, pp. 350–364. [Online]. Available: https://doi.org/10.1007/3-540-44618-4_26

[20] J. Goubault-Larrecq, "Prevision domains and convex powercones," in *FOSSACS 2008*. LNCS 4962, 2008, pp. 318–333. [Online]. Available: https://doi.org/10.1007/978-3-540-78499-9_23

[21] R. Tix, K. Keimel, and G. D. Plotkin, "Semantic domains for combining probability and non-determinism," *ENTCS*, vol. 222, pp. 3–99, 2009. [Online]. Available: https://doi.org/10.1016/j.entcs.2009.01.002

[22] R. Glabbeek, "The linear time - branching time spectrum i. the semantics of concrete, sequential processes," in *Handbook of Process Algebra*, J. A. Bergstra, A. Ponse, and S. Smolka, Eds. Amsterdam: Elsevier, 2001, pp. 3–99.

[23] R. Segala, "Modeling and verification of randomized distributed real-time systems," Ph.D. dissertation, MIT, 1995.

[24] ——, "A compositional trace-based semantics for probabilistic automata," in *Proc. CONCUR'95*, ser. LNCS, vol. 962. Springer, 1995, pp. 234–248.

[25] M. Bernardo, R. De Nicola, and M. Loreti, "Revisiting trace and testing equivalences for nondeterministic and probabilistic processes," *Logical Methods in Computer Science*, vol. 10(1:16), pp. 1–42, 2014.

[26] ——, "Relating strong behavioral equivalences for processes with nondeterminism and probabilities," *Theoretical Computer Science*, 2014.

[27] J. Rutten, "Universal coalgebra: A theory of systems," *Theoretical Computer Science*, vol. 249, pp. 3–80, 2000.

[28] B. Jacobs, *Introduction to Coalgebra: Towards Mathematics of States and Observation*, ser. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2016, vol. 59. [Online]. Available: https://doi.org/10.1017/CBO9781316823187

[29] I. Hasuo, B. Jacobs, and A. Sokolova, "Generic trace semantics via coinduction," *Logical Methods in Computer Science*, vol. 3, no. 4, 2007. [Online]. Available: https://doi.org/10.2168/LMCS-3(4:11)2007

[30] A. Silva, F. Bonchi, M. Bonsangue, and J. Rutten, "Generalizing the powerset construction, coalgebraically," in *Proc. FSTTCS 2010*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 8, 2010, pp. 272–283.

[31] M. Rabin, "Probabilistic automata," *Information and Control*, vol. 6, pp. 230–245, 1963.

[32] D. Varacca, "Probability, nondeterminism and concurrency: Two denotational models for probabilistic computation," Ph.D. dissertation, Univ. Aarhus, 2003, bRICS Dissertation Series, DS-03-14.

[33] W. Yi and K. Larsen, "Testing probabilistic and nondeterministic processes," in *Proc. PSTV'92*. North-Holland, 1992, pp. 47–61.

[34] B. Jonsson, C. Ho-Stuart, and W. Yi, "Testing and refinement for nondeterministic and probabilistic processes," in *Proc. FTRTFT'94*, ser. LNCS, vol. 863. Springer, 1994, pp. 418–430.

[35] Y. Deng, R. Glabbeek, M. Hennessy, C. Morgan, and C. Zhang, "Characterising testing preorders for finite probabilistic processes," in *Proc. LICS'07*. IEEE-CS Press, 2007, pp. 313–325.

[36] Y. Deng, R. Glabbeek, M. Hennessy, and C. Morgan, "Testing finitary probabilistic processes," in *Proc. CONCUR'09*, ser. LNCS, vol. 5710. Springer, 2009, pp. 274–288.

[37] R. Milner, *Communication and Concurrency*. Prentice Hall, 1989.

[38] D. Pous and D. Sangiorgi, "Enhancements of the bisimulation proof method," in *Advanced Topics in Bisimulation and Coinduction*, D. Sangiorgi and J. Rutten, Eds. Cambridge University Press, 2012.

[39] R. Segala, "Modeling and verification of randomized distributed real-time systems," Ph.D. dissertation, MIT, 1995.

[40] V. Castiglioni, "Trace and testing metrics on nondeterministic probabilistic processes," in *Proceedings Combined 25th International Workshop on Expressiveness in Concurrency and 15th Workshop on Structural Operational Semantics and 15th Workshop on Structural Operational Semantics, EXPRESS/SOS 2018, Beijing, China, September 3, 2018.*, 2018, pp. 19–36. [Online]. Available: https://doi.org/10.4204/EPTCS.276.4

[41] F. Bonchi, A. Sokolova, and V. Vignudelli, "The theory of traces for systems with nondeterminism and probability," *CoRR*, vol. abs/1808.00923v3, 2019. [Online]. Available: http://arxiv.org/abs/1808.00923

[42] T. Świrszcz, "Monadic functors and convexity," *Bull. Acad. Polon. Sci. Sér. Sci. Math. Astronom. Phys.*, vol. 22, pp. 39–42, 1974.

[43] Z. 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.

[44] E.-E. Doberkat, "Eilenberg-Moore algebras for stochastic relations," *Inform. and Comput.*, vol. 204, no. 12, pp. 1756–1781, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.ic.2006.09.001

[45] ——, "Erratum and addendum: Eilenberg-Moore algebras for stochastic relations [mr2277336]," *Inform. and Comput.*, vol. 206, no. 12, pp. 1476–1484, 2008. [Online]. Available: http://dx.doi.org/10.1016/j.ic.2008.08.002

[46] B. Jacobs, "Convexity, duality and effects," in *Theoretical computer science*, ser. IFIP Adv. Inf. Commun. Technol. Springer, Berlin, 2010, vol. 323, pp. 1–19. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15240-5_1

[47] M. Hyland, G. Plotkin, and J. Power, "Combining computational effects: Commutativity and sum," in *Foundations of Information Technology in the Era of Network and Mobile Computing*. Springer, 2002, pp. 474–484.

[48] A. Sokolova and H. Woracek, "Termination in convex sets of distributions," *Logical Methods in Computer Science*, vol. 14, no. 4, 2018.

[49] F. Dahlqvist, L. Parlant, and A. Silva, "Layer by layer - combining monads," in *Theoretical Aspects of Computing - ICTAC 2018 - 15th International Colloquium, Stellenbosch, South Africa, October 16-19, 2018, Proceedings*, ser. Lecture Notes in Computer Science, B. Fischer and T. Uustalu, Eds., vol. 11187. Springer, 2018, pp. 153–172. [Online]. Available: https://doi.org/10.1007/978-3-030-02508-3_9

[50] G. van Heerdt, J. Hsu, J. Ouaknine, and A. Silva, "Convex Language Semantics for Nondeterministic Probabilistic Automata," in *Theoretical Aspects of Computing - ICTAC 2018 - 15th International Colloquium, Stellenbosch, South Africa, October 16-19, 2018, Proceedings*, ser. Lecture Notes in Computer Science, B. Fischer and T. Uustalu, Eds., vol. 11187. Springer, 2018, pp. 472–492. [Online]. Available: https://doi.org/10.1007/978-3-030-02508-3

[51] S. Staton, "Relating coalgebraic notions of bisimulation," *Logical Methods in Computer Science*, vol. 7, no. 1, 2011.

[52] A. Kurz, "Logics for coalgebras and applications to computer science," Ph.D. dissertation, Ludwig-Maximilians-Universität München, 2000.

[53] U. Wolter, "On corelations, cokernels, and coequations," *Electronic Notes in Theoretical Computer Science*, vol. 33, 2000.

[54] F. Bartels, A. Sokolova, and E. d. Vink, "A hierarchy of probabilistic system types," *Theoretical Computer Science*, vol. 327, pp. 3–22, 2004.

[55] A. Sokolova, "Probabilistic systems coalgebraically: A survey," *Theor. Comput. Sci.*, vol. 412, no. 38, pp. 5095–5110, 2011.

[56] F. Bartels, "On generalised coinduction and probabilistic specification formats: distributive laws in coalgebraic modelling," Ph.D. dissertation, Vrije Universiteit, Amsterdam, 2004.

[57] B. Jacobs, A. Silva, and A. Sokolova, "Trace semantics via determinization," *J. Comput. Syst. Sci.*, vol. 81, no. 5, pp. 859–879, 2015.

[58] D. Turi and G. D. Plotkin, "Towards a mathematical operational semantics," in *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science, Warsaw, Poland, June 29 - July 2, 1997*. IEEE Computer Society, 1997, pp. 280–291. [Online]. Available: https://doi.org/10.1109/LICS.1997.614955

[59] B. Klin, "Bialgebras for structural operational semantics: An introduction," *Theor. Comput. Sci.*, vol. 412, no. 38, pp. 5043–5069, 2011. [Online]. Available: https://doi.org/10.1016/j.tcs.2011.03.023

[60] F. Bonchi, D. Petrisan, D. Pous, and J. Rot, "A general account of coinduction up-to," *Acta Inf.*, vol. 54, no. 2, pp. 127–190, 2017. [Online]. Available: https://doi.org/10.1007/s00236-016-0271-4

[61] F. Bonchi, M. M. Bonsangue, G. Caltais, J. Rutten, and A. Silva, "A coalgebraic view on decorated traces," *Mathematical Structures in Computer Science*, vol. 26, no. 7, pp. 1234–1268, 2016. [Online]. Available: https://doi.org/10.1017/S0960129514000449

[62] Y. Feng and L. Zhang, "When equivalence and bisimulation join forces in probabilistic automata," in *FM 2014: Formal Methods*, C. Jones, P. Pihlajasaari, and J. Sun, Eds. Cham: Springer International Publishing, 2014, pp. 247–262.

[63] P. Yang, D. N. Jansen, and L. Zhang, "Distribution-based bisimulation for labelled markov processes," in *FORMATS 2017*, 2017, pp. 170–186. [Online]. Available: https://doi.org/10.1007/978-3-319-65765-3_10

[64] B. Fischer and T. Uustalu, Eds., *Theoretical Aspects of Computing - ICTAC 2018 - 15th International Colloquium, Stellenbosch, South Africa, October 16-19, 2018, Proceedings*, ser. Lecture Notes in Computer Science, vol. 11187. Springer, 2018. [Online]. Available: https://doi.org/10.1007/978-3-030-02508-3