Initial algebras and final coalgebras consisting of nondeterministic finite trace strategies

Nathan Bowler [2]

Department of Mathematics
Universität Hamburg, Germany

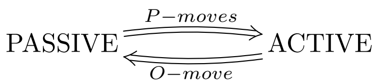
Paul Blain Levy [3]

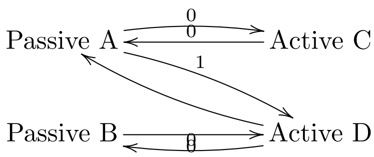
School of Computer Science
University of Birmingham, UK

Abstract

We study programs that perform I/O and finite nondeterministic choice, up to finite trace equivalence. For well-founded programs, we characterize which strategies (sets of traces) are definable, and axiomatize trace equivalence by means of commutativity between I/O and nondeterminism. This gives the set of strategies as an initial algebra for a polynomial endofunctor on semilattices. The strategies corresponding to non-well-founded programs constitute a final coalgebra for this functor. We also show corresponding results for countable nondeterminism.

Keywords: final coalgebra, nondeterministic strategies, trace, algebraic effects, semilattices





Consider the following (infinitary) imperative language:

$$\begin{array}{rcl} M, N & ::= & \mathsf{Age}(M_n)_{n \in \mathbb{N}} \\ & & \mid \mathsf{Happy}(M, N) \\ & & \mid \mathsf{Continue}(M) \\ & & \mid \mathsf{Bye} \mid M \texttt{ or } N \end{array}$$

The meaning is as follows.

[1] Thanks to everyone who should be thanked

[2] Email: Nathan.Bowler@uni-hamburg.de

[3] Email: P.B.Levy@cs.bham.ac.uk

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

- The command $\texttt{Age}(M_n)_{n\in\mathbb{N}}$ prints `What is your age?` and pauses. If the user then enters n, it executes M_n.
- The command $\texttt{Happy}(M,N)$ prints `Are you happy?` and pauses. If the user then enters `Yes` or `No`, it executes M or N respectively.
- The command $\texttt{Continue}(M)$ prints `Continue?` and pauses. If the user then enters `Yes`, it executes M.
- The command `Bye` prints `Goodbye` and pauses. No further input is possible.
- The command M `or` N nondeterministically chooses to execute M or N.

Any command has a set of traces, which are alternating sequences of outputs and inputs called *plays*. For example:

```
Are you happy?
Yes
What is your age?
```

Two commands with the same traces are *trace equivalent*. The following questions naturally arise:

(i) Given a set P of plays, under what conditions is P the trace set of some command?

(ii) Can we give an axiomatic theory of trace equivalence?

This paper's main contribution is to answer these questions. The answer to question ii is surprisingly simple: we take the ordinary theory of `or` (commutativity, associativity and idempotency), together with the fact that each I/O operation *commutes* with `or`. For example:

$$\texttt{Age}(M_n)_{n\in\mathbb{N}} \texttt{ or } \texttt{Age}(M'_n)_{n\in\mathbb{N}} = \texttt{Age}(M_n \texttt{ or } M'_n)_{n\in\mathbb{N}}$$

We give our results not only for the language above but also for some variations, as we shall now explain. The language has two parts—I/O and nondeterminism—and each can be varied.

(i) The I/O part is determined by a *signature*, a collection of operations each with a specified arity—a set of argument indices. The language above has four I/O operations—Age, Happy, Continue and Bye—of respective arity $\mathbb{N}$, $\{\mathsf{Yes},\mathsf{No}\}$, $\{\mathsf{Yes}\}$ and $\emptyset$. Our results apply no matter what I/O signature is used to generate the language.

(ii) We vary the nondeterministic part as follows.
   - We consider the command $\texttt{choose}\ (M_n)_{n\in\mathbb{N}}$, which nondeterministically chooses $n \in \mathbb{N}$ and then executes M_n.
   - More generally, we consider $\texttt{choose}\ (M_i)_{i\in I}$ where I may be uncountable.

The significance of these results is shown by their connection to several areas of semantics.

Effects and monads. I/O operations and nondeterministic choice are examples of *computational effects*. A collection of effects is often described by a monad on **Set** [17], which can sometimes be presented by a simple theory [18]. For each of our variations, our results give rise to a monad on **Set**, corresponding to programs modulo trace equivalence, which is moreover a *tensor* of the monads for I/O and nondeterminism [1,4,5,11].

Coalgebraic traces. An algebra for our theory may be seen as an algebra for the I/O signature in the category of semilattices, or equivalently as an algebra for a *polynomial endofunctor* on that category. The definable sets of plays form an initial such algebra, corresponding to the fact that programs are well-founded. The second part of the paper treats another variation: *non-well-founded* programs. Such programs may have infinite traces, but we ignore these and treat only the finite traces. We show that the definable sets of plays form a *final coalgebra* for the same endofunctor. This gives a coalgebraic account of finite trace semantics. Although several coalgebraic accounts of traces have appeared [8,13,14], the novelty of ours is that traces include both output and input actions.

Game semantics. A program in the language above may be seen as playing a game. At any time, it is either in an *active position*, i.e. executing, or in a *passive position*, i.e. paused. There is a passive position corresponding to each operation, but only one active position, where play begins. By contrast, the games used in game semantics may have many active and many passive positions [16]. A different terminology is used: an output action is called a *P-move* (for "Proponent") and an input action an *O-move* (for "Opponent"). Nonetheless, where finite traces are studied, the same notions of nondeterministic strategies [6,7] may be used, and our results characterize these strategies for general games.

Notation. Given a set X, we write $\mathcal{P}^+_{\mathsf{f}} X$ for the set of finite inhabited subsets, $\mathcal{P}^+_{\mathsf{c}} X$ for the set of countable inhabited subsets, and $\mathcal{P}^+ X$ for the set of inhabited subsets.

Acknowledgements

We thank G. Plotkin and V. Vignudelli for helpful discussion. Theorem 1.18 was independently proved by Plotkin.

1 Finite nondeterminism

In this section, we define the finitely nondeterministic language for a signature, define bisimulation and trace equivalence of programs, characterize definable strategies, axiomatize bisimilarity and trace equivalence. Lastly we directly describe the semilattice of strategies as an initial algebra without needing to mention the language.

1.1 Language

Definition 1.1 A *signature* consists of a set K of *operations*, and for each operation $k \in K$, a set $\mathsf{Ar}(k)$ of *argument indices*.

For the sequel, we fix a signature $S = (\mathsf{Ar}(k))_{k \in K}$.

Definition 1.2 The set comm of *commands* is given inductively by the grammar

$$M ::= \ \mathtt{In}\, k(M_i)_{i \in \mathsf{Ar}(k)} \ \mid \ M \ \mathtt{or} \ M$$

A command without $\mathtt{or}$ is *deterministic*.

We give operational semantics via a transition system, cf. [19].

Definition 1.3

(i) For $k \in K$, a *k-passive state* (representing a paused program) is a tuple $x = (M_i)_{i \in \mathsf{Ar}(k)}$ of commands. For $i \in \mathsf{Ar}(k)$ we write $x@i \stackrel{\text{def}}{=} M_i$.

(ii) We define a relation $M \stackrel{k}{\Rightarrow} x$, where M is a command, $k \in K$ and x is a k-passive state, meaning that M may output k and then be in state x. The relation is defined inductively by the rules

$$\frac{}{\mathtt{In}\, k(M_i)_{i \in \mathsf{Ar}(k)} \stackrel{k}{\Rightarrow} (M_i)_{i \in \mathsf{Ar}(k)}} \qquad \frac{M \stackrel{k}{\Rightarrow} x}{M \ \mathtt{or} \ N \stackrel{k}{\Rightarrow} x} \qquad \frac{N \stackrel{k}{\Rightarrow} x}{M \ \mathtt{or} \ N \stackrel{k}{\Rightarrow} x}$$

(iii) We define $\zeta \colon \mathsf{comm} \longrightarrow \mathcal{P} \sum_{k \in K} \prod_{i \in \mathsf{Ar}(k)} \mathsf{comm}$ to send M to the set of all (k, x) such that $M \stackrel{k}{\Rightarrow} x$.

The transition system has the following properties.

Proposition 1.4 *Let M be a command.*

(i) *(Finite nondeterminism and totality) The set ζM is finite and inhabited.*

(ii) *(Well-foundedness) There is no infinite sequence*

$$M \stackrel{k_0}{\Rightarrow} x_0 \quad x_0@i_0 \stackrel{k_1}{\Rightarrow} x_1 \quad x_1@i_1 \stackrel{k_2}{\Rightarrow} x_2 \quad \cdots$$

Proof. Induction on M. □

1.2 Bisimulation

Definition 1.5

(i) A *passive relation* $\mathcal{R}$ associates to each $k \in K$ a binary relation $\mathcal{R}_k$ on k-passive states.

(ii) A passive relation $\mathcal{R}$ is a *bisimulation* when for $x \,\mathcal{R}_k\, x'$ and $i \in \mathsf{Ar}(k)$, if $x@i \stackrel{l}{\Rightarrow} y$ then there is an l-passive state y' such that $x'@i \stackrel{l}{\Rightarrow} y'$ and $y \,\mathcal{R}_l\, y'$, and *vice versa*.

(iii) Commands M, M' are *bisimilar* when there exists a bisimulation $\mathcal{R}$ such that if $M \stackrel{k}{\Rightarrow} y$ then there is a k-passive state y' such that $M' \stackrel{k}{\Rightarrow} y'$ and $y \,\mathcal{R}_k\, y'$, and *vice versa*.

Axiomatizing bisimilarity is easy.

Proposition 1.6 *The least congruence $\equiv$ on commands satisfying*

$$\begin{aligned} M \texttt{ or } N &\equiv N \texttt{ or } M \\ (M \texttt{ or } N) \texttt{ or } P &\equiv M \texttt{ or } (N \texttt{ or } P) \\ M \texttt{ or } M &\equiv M \end{aligned}$$

is bisimilarity.

Proof. In Appendix. □

1.3 Traces and Strategies

Definition 1.7

(i) A *play* is a sequence $k_0, i_0, k_1, i_1, \ldots$ where $k_r \in K$ and $i_r \in \mathsf{Ar}(k_r)$. It is *active-ending*, *passive-ending* or *infinite* according as its length is even, odd or infinite.

(ii) A *prerequisite* of a play is a passive-ending strict prefix.

Intuitively, at the end of a passive-ending play, execution is paused and waiting for input.

Definition 1.8 Let $\zeta \colon X \longrightarrow \mathcal{P} \sum_{k \in K} \prod_{i \in \mathsf{Ar}(k)} X$ be a transition system. (For example, our language, with $X = \mathsf{comm}$.) Let $M \in X$. A play $k_0, i_0, k_1, i_1, \ldots$ is a *trace* of M when there exists a sequence

$$M \overset{k_0}{\Rightarrow} x_0 \quad x_0 @ i_0 \overset{k_1}{\Rightarrow} x_1 \quad x_1 @ i_1 \overset{k_2}{\Rightarrow} x_2 \quad \cdots$$

Definition 1.9

(i) A *nondeterministic strategy* (in the sense of finite traces) is a set σ of passive-ending plays, such that if $s \in \sigma$ then so are its prerequisites.

(ii) Let σ be a nondeterministic strategy. We write σ^{AE} for the set of active-ending plays whose prerequisites are in σ, i.e. for the set $\{\varepsilon\} \cup \{ski \mid sk \in \sigma, i \in \mathsf{Ar}(k)\}$.

In particular, for a transition system $\zeta \colon X \longrightarrow \mathcal{P} \sum_{k \in K} \prod_{i \in \mathsf{Ar}(k)} X$ and any $M \in X$, the set of passive-ending traces of M forms a strategy, written $\mathsf{Traces}\, M$. The set of active-ending traces is $(\mathsf{Traces}\, M)^{\mathsf{AE}}$.

If M is a command, we may also describe $\mathsf{Traces}\, M$ compositionally, as follows. We use the notation $x.t$ to mean x prepended to the sequence t.

Definition 1.10 Let $k \in K$ and let $(\sigma_i)_{i \in \mathsf{Ar}(k)}$ be a family of nondeterministic strategies. The nondeterministic strategy $\mathsf{In}\, k(\sigma_i)_{i \in I}$ is the set of plays $k.t$, where either $t = \varepsilon$ or $t = i.s$ for some $i \in \mathsf{Ar}(k)$ and $s \in \sigma_i$.

Proposition 1.11 *We give* $\mathsf{Traces}\, M$ *compositionally:*

$$\begin{aligned} \mathsf{Traces}\, \mathsf{In}\, k(M_i)_{i \in \mathsf{Ar}(k)} &= \mathsf{In}\, k(\mathsf{Traces}\, M_i)_{i \in \mathsf{Ar}(k)} \\ \mathsf{Traces}\, (M \texttt{ or } N) &= \mathsf{Traces}\, M \cup \mathsf{Traces}\, N \end{aligned}$$

Hence traces respect $\equiv$, and we write $\mathsf{Traces}\, A \overset{\text{def}}{=} \mathsf{Traces}\, M$ when $A = [M]_{\equiv}$. As usual, bisimilarity is finer than trace equivalence:

Proposition 1.12 *$M \sim N$ implies $\mathsf{Traces}\, M = \mathsf{Traces}\, N$, but not conversely.*

Proof. The commands $a.(b \texttt{ or } c)$ and $a.b \texttt{ or } a.c$, where a is a unary operation and b, c are constants (nullary operations), are trace equivalent but not bisimilar. □

We may also decompose strategies, as follows.

Definition 1.13 Let σ be a strategy. We write $\mathsf{Init}\, \sigma$ for the set of $k \in K$ such that $k \in \sigma$. For each such k and each $i \in \mathsf{Ar}(k)$, we define the strategy $\sigma / ki \overset{\text{def}}{=} \{s \mid k.i.s \in \sigma\}$.

1.4 Definability

We now consider our first question: which nondeterministic strategies are of the form $\mathsf{Traces}\, M$ for some command M? To answer this, we list several conditions.

Definition 1.14 Let σ be a nondeterministic strategy.

(i) A *response* to a play $s \in \sigma^{\mathsf{AE}}$ is $k \in K$ such that $sk \in \sigma$. The set of responses is written $\mathsf{Resp}(\sigma, s)$.
(ii) σ is a *tree* when every $s \in \sigma^{\mathsf{AE}}$ has a unique response.
(iii) σ is *total* when every $s \in \sigma^{\mathsf{AE}}$ has at least one response.
(iv) σ is *deterministic*, or a *partial tree*, when every $s \in \sigma^{\mathsf{AE}}$ has at most one response.
(v) σ is *finitely nondeterministic* when every $s \in \sigma^{\mathsf{AE}}$ has only finitely many responses. (Automatic if K is finite.)
(vi) σ is *anti-König* when it is finitely nondeterministic and there is no infinite play whose prerequisites are all in σ. A tree or partial tree with the latter property is also called *well-founded*.

The following is obvious.

Proposition 1.15 *The mapping $M \mapsto \mathsf{Traces}\, M$ is a bijection from deterministic commands to well-founded trees.*

Proposition 1.16 *For a nondeterministic strategy σ, the following are equivalent.*

- $\sigma = \mathsf{Traces}\, M$ *for some command M.*
- σ *is total and anti-König.*

Proof. ($\Rightarrow$) is proved by induction on M, or from the properties of the transition system (Proposition 1.4). For ($\Leftarrow$), see the Appendix. □

1.5 Axiomatizing trace equivalence

Definition 1.17 Let $\equiv_{\mathsf{c}}$ be the least congruence on commands that contains $\equiv$ and also

$$\texttt{In}\, k(M_i \texttt{ or } N_i)_{i \in \mathsf{Ar}(k)} \equiv_{\mathsf{c}} \texttt{In}\, k(M_i)_{i \in \mathsf{Ar}(k)} \texttt{ or } \texttt{In}\, k(N_i)_{i \in \mathsf{Ar}(k)} \tag{1}$$

For $A = [M]_{\equiv}$ and $B = [N]_{\equiv}$, we write $A =_{\mathsf{c}} B$ when $M \equiv_{\mathsf{c}} N$.

The equation (1) is called *commutativity between I/O and nondeterminism.*

Theorem 1.18 *For commands M and N, we have $M \equiv_{\mathsf{c}} N$ iff $\mathsf{Traces}\, M = \mathsf{Traces}\, N$.*

Proof. In Appendix. □

1.6 Semilattices and algebras

The algebraic view of binary nondeterminism is as follows.

Definition 1.19

(i) A *semilattice* is a set X equipped with a binary operation $\vee$ that is commutative, associative and idempotent.
(ii) A semilattice *homorphism* $(X, \vee) \longrightarrow (Y, \vee')$ is a function $f \colon X \longrightarrow Y$ such that $f(x \vee y) = f(x) \vee f(y)$.

Proposition 1.20 *A semilattice may equivalently be described as a poset with all binary joins:*

- *we write $x \leqslant y$ when $x \vee y = y$*
- *conversely, $x \vee y$ is the join of x and y.*

A semilattice homomorphism is a function that preserves binary joins and hence is monotone.

Next we give the algebraic view of the I/O operations provided by our signature S.

Definition 1.21

(i) An S-*algebra* consists of a set X and, for each $k \in K$, a function $[\![k]\!] \colon \prod_{i \in \mathsf{Ar}(k)} X \longrightarrow X$.

(ii) An S-algebra *homomorphism* $(X, (\llbracket k \rrbracket)_{k \in K}) \longrightarrow (Y, (\llbracket k \rrbracket')_{k \in K})$ is a function $f\colon X \longrightarrow Y$ satisfying $f(\llbracket k \rrbracket (x_i)_{i \in \mathsf{Ar}(k)}) = \llbracket k \rrbracket'(f x_i)_{i \in \mathsf{Ar}(k)}$ for all $k \in K$.

This leads to a standard result:

Proposition 1.22 *The set of well-founded trees with $(\mathsf{In}\, k)_{k \in K}$ is an initial S-algebra.*

Our aim is to combine nondeterminism and I/O in a similar way. We generalize Definition 1.21 as follows.

Definition 1.23 Let $\mathcal{C}$ be a category with products.

(i) An *S-algebra* in $\mathcal{C}$ consists of $X \in \mathcal{C}$ and, for each $k \in K$, a morphism $\llbracket k \rrbracket \colon \prod_{i \in \mathsf{Ar}(k)} X \longrightarrow X$.

(ii) An S-algebra *homomorphism* $(X, (\llbracket k \rrbracket)_{k \in K}) \longrightarrow (Y, (\llbracket k \rrbracket')_{k \in K})$ is a morphism $f\colon X \longrightarrow Y$ such that

$$\begin{array}{ccc} \prod_{i \in \mathsf{Ar}(k)} X & \xrightarrow{\prod_{i \in \mathsf{Ar}(k)} f} & \prod_{i \in \mathsf{Ar}(k)} Y \\ {\scriptstyle \llbracket k \rrbracket}\downarrow & & \downarrow{\scriptstyle \llbracket k \rrbracket'} \\ X & \xrightarrow[f]{} & Y \end{array}$$

commutes for all $k \in K$.

Unpacking this definition, an S-algebra in semilattices consists of a semilattice $(X, \vee)$ and a family $\llbracket k \rrbracket_{k \in K}$ of functions $\llbracket k \rrbracket \colon X^{\mathsf{Ar}(k)} \longrightarrow X$ that satisfy

$$\llbracket k \rrbracket (x_i \vee y_i)_{i \in \mathsf{Ar}(k)} \;=\; \llbracket k \rrbracket (x_i)_{i \in \mathsf{Ar}(k)} \vee \llbracket k \rrbracket (y_i)_{i \in \mathsf{Ar}(k)}$$

A homomorphism is a function preserving $\vee$ and $\llbracket k \rrbracket$ for all $k \in K$. We therefore have an initial S-algebra in semilattices, viz. the set of commands modulo $\equiv_{\mathsf{c}}$, with $\mathtt{or}$ and $(\mathtt{In}\, k)_{k \in K}$. In view of Propositions 1.16 and 1.18, it is isomorphic via $M \mapsto \mathsf{Traces}\, M$ to the semilattice of total anti-König strategies, ordered by inclusion, with $(\mathsf{In}\, k)_{k \in K}$. To summarize:

Theorem 1.24 *The set of total anti-König strategies, ordered by inclusion, with $(\mathsf{In}\, k)_{k \in K}$, is an initial S-algebra in semilattices.*

2 Countable Nondeterminism

This section adapts all our results from finite to countable nondeterminism. There are significant changes in the condition on strategies, and the proof method for completeness of the theory.

2.1 Language

We now extend our language to include countable nondeterministic choice:

$$M ::= \quad \mathtt{In}\, k(M_i)_{i \in \mathsf{Ar}(k)} \;\mid\; M \;\mathtt{or}\; M \;\mid\; \mathtt{choose}\ (M_n)_{n \in \mathbb{N}}$$

We add to the inductive definition of $\Rightarrow$ the rule

$$\frac{M_n \stackrel{k}{\Rightarrow} x}{\mathtt{choose}\ (M_n)_{n \in \mathbb{N}} \stackrel{k}{\Rightarrow} x}$$

By comparison with Proposition 1.4, the set ζM is now *countable* and inhabited. The system remains well-founded.

2.2 Bisimulation

Axiomatizing bisimilarity requires additional equations:

Definition 2.1 Let $\equiv$ be the least congruence on commands satisfying

$$\begin{aligned} M \texttt{ or } N &\equiv N \texttt{ or } M \\ (M \texttt{ or } N) \texttt{ or } P &\equiv M \texttt{ or } (N \texttt{ or } P) \\ M \texttt{ or } M &\equiv M \\ M \texttt{ or choose } (M_n)_{n\in\mathbb{N}} &\equiv \texttt{choose } (M \texttt{ or } M_n)_{n\in\mathbb{N}} \\ \texttt{choose } (M)_{n\in\mathbb{N}} &\equiv M \\ \texttt{choose } (M_n)_{n\in\mathbb{N}} &\equiv \texttt{choose } (M_n)_{n\in\mathbb{N}} \texttt{ or } M_n \end{aligned}$$

A semilattice with an ω-ary operation satisfying these equations is called an *ω-semilattice*. This corresponds, as in Proposition 1.20, to a poset in which every family $(a_i)_{i\in\mathbb{N}}$ has a supremum. A homomorphism of ω-semilattices must preserve these suprema. We prove that $\equiv$ coincides with bisimilarity as before.

2.3 Definability

We again want to characterize those nondeterministic strategies that are of the form $\mathsf{Traces}\, M$. But, as has often been observed, we cannot have a condition similar to the anti-König one. For consider the command $M \stackrel{\text{def}}{=} \texttt{choose } (a^n.b)_{n\in\mathbb{N}}$, where a is unary and b is constant: the infinite play a^ω has all its prerequisites in $\mathsf{Traces}\, M$. Accordingly, we consider instead the following conditions.

Definition 2.2 Let σ be a nondeterministic strategy.

(i) σ is *countably nondeterministic* when every $s \in \sigma^{\mathsf{AE}}$ has only countably many responses. (Automatic if K is countable.)

(ii) σ is *well-foundedly total* when for every $s \in \sigma^{\mathsf{AE}}$, there is a well-founded tree τ such that $st \in \sigma$ for all $t \in \tau$.

To understand the latter condition, we first define, for any command M, its *left determinization* $\mathsf{LD}(M)$.

$$\begin{aligned} \mathsf{LD}(\texttt{In}\, k(M_i)_{i\in\mathsf{Ar}(k)}) &\stackrel{\text{def}}{=} \texttt{In}\, k(\mathsf{LD}(M_i))_{i\in\mathsf{Ar}(k)} \\ \mathsf{LD}(M \texttt{ or } N) &\stackrel{\text{def}}{=} \mathsf{LD}(M) \\ \mathsf{LD}(\texttt{choose } (M_n)_{n\in\mathbb{N}}) &\stackrel{\text{def}}{=} \mathsf{LD}(M_0) \end{aligned}$$

We argue that $\mathsf{Traces}\, M$ is well-foundedly total: if $s \in (\mathsf{Traces}\, M)^{\mathsf{AE}}$, then there is some execution of M that performs s and ends in a command N, and then $\mathsf{Traces}\,\mathsf{LD}(N)$ is a well-founded tree with the required property. We also see that $\mathsf{Traces}\, M$ is countably nondeterministic by the same argument as before.

Proposition 2.3 *For any strategy σ, the following are equivalent.*

- $\sigma = \mathsf{Traces}\, M$ *for some command M.*
- σ *is well-foundedly total and countably nondeterministic.*

Proof. We have just shown ($\Rightarrow$). For ($\Leftarrow$) see the Appendix. □

2.4 Axiomatizing Trace Equivalence

We define $\equiv_{\mathsf{c}}$ as before: each I/O operation commutes with nondeterministic choice.

Definition 2.4 Let $\equiv_{\mathsf{c}}$ be the least congruence on commands that contains $\equiv$ and also

$$\begin{aligned} \texttt{In}\, k(M_i \texttt{ or } N_i)_{i\in\mathsf{Ar}(k)} &\equiv_{\mathsf{c}} \texttt{In}\, k(M_i)_{i\in\mathsf{Ar}(k)} \texttt{ or } \texttt{In}\, k(N_i)_{i\in\mathsf{Ar}(k)} \\ \texttt{In}\, k(\texttt{choose } (M_{i,n})_{n\in\mathbb{N}})_{i\in\mathsf{Ar}(k)} &\equiv_{\mathsf{c}} \texttt{choose } (\texttt{In}\, k(M_{i,n})_{i\in\mathsf{Ar}(k)})_{n\in\mathbb{N}} \end{aligned}$$

For $A = [M]_\equiv$ and $B = [N]_\equiv$, we write $A =_{\mathsf{c}} B$ when $M \equiv_{\mathsf{c}} N$.

Theorem 2.5 *For commands M and N, we have $M \equiv_{\mathsf{c}} N$ iff $\mathsf{Traces}\, M = \mathsf{Traces}\, N$.*

The proof is in the Appendix. It differs from the one used for the finite case because trace normal form is not available.

2.5 Infinite nondeterminism and algebras

As in Section 1.6, we conclude the following:

Theorem 2.6 *The set of well-foundedly total and countably nondeterministic strategies, ordered by inclusion, with* $(\mathsf{In}\, k)_{k \in K}$*, is an initial* S*-algebra in* ω*-semilattices.*

The countable nondeterminism condition is automatic if K is countable. For a general signature, it may be removed as follows. Say that an *almost complete semilattice* is a (join) semilattice where every inhabited subset has a least upper bound. A *homomorphism* is a function that preserves these least upper bounds.

Theorem 2.7 *The set of well-foundedly total strategies, ordered by inclusion, with* $(\mathsf{In}\, k)_{k \in K}$*, is an initial* S*-algebra in almost complete semilattices.*

Proof. In Appendix. □

We may also adapt Theorem 2.6 to nondeterminism bounded by a regular uncountable cardinal.

3 From a signature to an endofunctor

In this section, we present the set of strategies as an initial algebra or a final coalgebra for an endofunctor. We also see how the latter gives rise to a notion of bisimulation that relates sets of states, giving a proof method for trace equivalence. While this appears to be new, numerous authors have considered notions of bisimulation that relate probability distribution, e.g. [2,3,9,10,20]

3.1 Initial algebras

Recall that Definition 1.23 applies to any category $\mathcal{C}$ with products. If $\mathcal{C}$ also has coproducts, written $\bigoplus$, then S-algebras in $\mathcal{C}$ may be described as algebras for the endofunctor $\bigoplus_{k \in K} \prod_{i \in \mathsf{Ar}(k)}$. Each of our categories—semilattices, ω-semilattices and almost complete semilattices—has coproducts with a simple explicit description.

Proposition 3.1

(i) *A coproduct* $\bigoplus_{j \in J} A_j$ *of semilattices is given by the set* $\sum_{U \in \mathcal{P}_{\mathsf{f}}^{+} J} \prod_{j \in U} A_j$ *with* $(U, (a_j)_{j \in U}) \leqslant (V, (b_j)_{j \in V})$ *when* $U \subseteq V$ *and* $a_j \leqslant b_j$ *for all* $j \in U$*. For* $j \in J$*, the* j*th embedding* $e_j \colon A_j \longrightarrow \bigoplus_{j \in J} A_j$ *sends* $a \mapsto (\{j\}, (a)_j)$*. The cotuple of a family of homomorphisms* $(f_j \colon A_j \longrightarrow B)_{j \in J}$ *sends* $(U, (a_j)_{j \in U}) \mapsto \bigvee_{j \in U} f_j(a_j)$*.*

(ii) *Likewise for* ω*-semilattices, using* $\mathcal{P}_{\mathsf{c}}^{+} J$*.*

(iii) *Likewise for almost complete semilattices, using* $\mathcal{P}^{+} J$*.*

Let us reformulate Theorem 1.24 in these terms.

Theorem 3.2 *The set of total anti-König strategies, ordered by inclusion with structure*

$$(U, (\sigma_{k,i})_{k \in K, i \in \mathsf{Ar}(k)}) \mapsto \bigcup_{k \in U} \mathtt{In}\, k(\sigma_{k,i})_{i \in \mathsf{Ar}(k)}$$

forms an initial algebra for $\bigoplus_{k \in K} \prod_{i \in \mathsf{Ar}(k)}$ *on semilattices.*

Lambek's lemma says that the structure of an initial algebra an isomorphism. In this case its inverse sends σ to $(\mathsf{Init}\, \sigma, (\sigma / ki)_{k \in \mathsf{Init}\, \sigma, i \in \mathsf{Ar}(k)})$. Likewise for Theorems 2.6–2.7.

3.2 Final coalgebras

We now consider non-well-founded total systems. We treat only the finitely nondeterministic case, but the countably nondeterministic and unconstrained cases are similar. As usual, our first question is definability of strategies. We write $\mathsf{Strat}_{\mathsf{f}}^{+}$ for the set of finitely nondeterministic, total strategies.

Proposition 3.3 *For a strategy* σ *the following are equivalent.*

- $\sigma = \mathsf{Traces}\, M$ *for some element* M *of a transition system* $\zeta \colon X \longrightarrow \mathcal{P}_{\mathsf{f}} \sum_{k \in K} \prod_{i \in \mathsf{Ar}(k)} X$*.*
- σ *is finitely nondeterministic and total.*

Proof. $(\Rightarrow)$ is evident. For $(\Leftarrow)$, we put $X = \mathsf{Strat}^+_{\mathsf{f}}$ and $\zeta\colon \sigma \mapsto \{(k, (\sigma/ki)_{i\in\mathsf{Ar}(k)}) \mid k \in \mathsf{Init}\,\sigma\}$, where $\Psi\sigma = (U, (\sigma_{k,i})_{k\in U, i\in\mathsf{Ar}(k)})$. For any passive-ending play s, we show $s \in \mathsf{Traces}\,\sigma$ iff $s \in \sigma$, by induction on s, separating the cases $s = (k)$ and $s = k.i.s'$. □

Proposition 3.4 *The function* $\Psi\colon \mathsf{Strat}^+_{\mathsf{f}} \longrightarrow \bigoplus_{k\in K}\prod_{i\in\mathsf{Ar}(k)}\mathsf{Strat}^+_{\mathsf{f}}$ *sending* σ *to* $(\mathsf{Init}\,\sigma, (\sigma/ki)_{k\in\mathsf{Init}\,\sigma, i\in\mathsf{Ar}(k)})$ *is a semilattice isomorphism.*

We use this to characterize traces for a transition system, in terms of the following notions.

Definition 3.5 For each $j \in J$, let X_j be a set, A_j a semilattice and $f_j\colon X_j \longrightarrow A_j$ a function. Then we write $\Omega_{j\in J} f_j\colon \mathcal{P}^+_{\mathsf{f}}\sum_{j\in J} X_j \longrightarrow \bigoplus_{j\in J} A_j$ for the unique homomorphism h such that

$$\begin{array}{ccc} X_j & \xrightarrow{f_j} & A_j \\ {\scriptstyle\{\mathsf{in}_j\,-\}}\downarrow & & \downarrow{\scriptstyle e_j} \\ \mathcal{P}^+_{\mathsf{f}}\sum_{j\in J} X_j & \xrightarrow[h]{} & \bigoplus_{j\in J} A_j \end{array}$$

commutes for all $j \in J$. Explicitly it sends R to $(U, (y_j)_{j\in U})$ where

$$\begin{aligned} U &= \{j \in J \mid \exists x \in X_j \mid \mathsf{in}_j\, x \in R\} \\ y_j &= \bigvee_{x\in X_j \mid \mathsf{in}_j\, x\in R} f_j(x_j) \quad \text{for } j \in U. \end{aligned}$$

Definition 3.6 Let (X, ζ) be a transition system i.e. $\mathcal{P}_{\mathsf{f}}\sum_{k\in K}\prod_{i\in\mathsf{Ar}(k)}$-coalgebra and (A, ξ) a $\bigoplus_{k\in K}\prod_{i\in\mathsf{Ar}(k)}$-coalgebra. A *map* $h\colon (X,\zeta) \longrightarrow (A, \xi)$ is a function $X \longrightarrow A$ such that

$$\begin{array}{ccc} X & \xrightarrow{h} & A \\ {\scriptstyle\zeta}\downarrow & & \downarrow{\scriptstyle\xi} \\ \mathcal{P}_{\mathsf{f}}\sum_{k\in K}\prod_{i\in\mathsf{Ar}(k)} X & \xrightarrow[\Omega_{j\in J}\prod_{i\in\mathsf{Ar}(k)} h]{} & \bigoplus_{k\in K}\prod_{i\in\mathsf{Ar}(k)} A \end{array}$$

commutes.

We note that such a map can be precomposed with a coalgebra morphism $(X', \zeta') \longrightarrow (X, \zeta)$, or postcomposed with a coalgebra morphism $(A, \xi) \longrightarrow (A', \xi')$, by function composition.

Theorem 3.7 *Let* $\zeta\colon X \longrightarrow \mathcal{P}_{\mathsf{f}}\sum_{k\in K}\prod_{i\in\mathsf{Ar}(k)} X$ *be a transition system. Then* $M \mapsto \mathsf{Traces}\,M$ *is the unique map from* (X, ζ) *to* $(\mathsf{Strat}^+_{\mathsf{f}}, \Psi)$.

Proof. In Appendix. □

A similar construction gives the coalgebraic counterpart of Theorem 3.2.

Theorem 3.8 $(\mathsf{Strat}^+_{\mathsf{f}}, \Psi)$ *is a final* $\bigoplus_{k\in K}\prod_{i\in\mathsf{Ar}(k)}$*-coalgebra.*

Proof. Let (A, ζ) be a $\bigoplus_{k\in K}\prod_{i\in\mathsf{Ar}(k)}$-coalgebra. For $M \in A$, we write $\mathsf{Traces}\,M$ for the set of passive-ending plays that are *traces* of A, suitably defined. Then $M \mapsto \mathsf{Traces}\,M$ is the required coalgebra morphism to $(\mathsf{Strat}^+_{\mathsf{f}}, \Psi)$. The details are in the Appendix. □

3.3 Determinization and Bisimulation

The results we have seen give rise to a determinization process that may be used to establish when states are trace equivalent. This resembles the account of determinization in [13], and indeed both are instances of the general framework in [12].

Proposition 3.9 *Let* $\zeta\colon X \longrightarrow \mathcal{P}_{\mathsf{f}}\sum_{k\in K}\prod_{i\in\mathsf{Ar}(k)} X$ *be a transition system. Then there is a unique semilattice homomorphism* $\hat{\zeta}\colon \mathcal{P}^+_{\mathsf{f}} X \longrightarrow \bigoplus_{k\in K}\prod_{i\in\mathsf{Ar}(k)}\mathcal{P}^+_{\mathsf{f}} X$ *such that* $\{-\}\colon (X, \zeta) \longrightarrow (\mathcal{P}^+_{\mathsf{f}} X, \hat{\zeta})$ *is a map. Explicitly it sends* R *to* $(U, (S_k)_{k\in U})$*, where*

$$\begin{aligned} U &= \{k \in K \mid \exists M \in R.\, M \overset{k}{\Rightarrow} y\} \\ S_k@i &= \{y@i \mid \exists M \in R.\, M \overset{k}{\Rightarrow} y\} \quad \textit{for } k \in U \textit{ and } i \in \mathsf{Ar}(k). \end{aligned}$$

Proof. It is the Kleisli extension of

$$X \xrightarrow{\zeta} \mathcal{P}_{\mathsf{f}} \sum_{k\in K} \prod_{i\in \mathsf{Ar}(k)} X \xrightarrow{(\mathcal{P}^+_{\mathsf{f}})^{\sum}_{k\in K} \prod_{i\in\mathsf{Ar}(k)}\{-\}} \bigoplus_{k\in K} \prod_{i\in \mathsf{Ar}(k)} \mathcal{P}^+_{\mathsf{f}} X$$

. ◻

It follows from Theorems 3.7–3.8 that $\mathsf{Traces}\, M = \mathsf{Traces}\,\{M\}$. This reduces the problem of establishing trace equivalence in a transition system to that of establishing trace equivalence in a $\bigoplus_{k\in K} \prod_{i\in \mathsf{Ar}(k)}$-coalgebra. We now give a bisimulation method for the latter.

Definition 3.10

(i) Let A and B be semilattices. A *semilattice relation* $A \xrightarrow{\mathcal{R}} B$ is a relation such that $x\,\mathcal{R}\,x'$ and $y\,\mathcal{R}\,y'$ implies $x \vee y\,\mathcal{R}\,x' \vee y'$.

(ii) For each $j \in J$ let $A_j \xrightarrow{\mathcal{R}} B_j$ be a semilattice relation. The semilattice relation $\prod_{j\in J} A_j \xrightarrow{\prod_{j\in j}\mathcal{R}} \prod_{j\in J} B_j$ relates $(x_j)_{j\in J}$ to $(x'_j)_{j\in J}$ when $x_j\,\mathcal{R}\,x'_j$ for all $j \in J$.

(iii) For each $j \in J$ let $A_j \xrightarrow{\mathcal{R}} B_j$ be a semilattice relation. The semilattice relation $\bigoplus_{j\in J} A_j \xrightarrow{\bigoplus_{j\in j}\mathcal{R}} \bigoplus_{j\in J} B_j$ relates $(U, (y_j)_{j\in U})$ to $(U', (y_j)_{j\in U'})$ when $U = U'$ and $y_j\,\mathcal{R}\,y'_j$ for all $j \in U$.

Definition 3.11 Let (A, ζ) and (A', ζ') be $\bigoplus_{k\in K} \prod_{i\in \mathsf{Ar}(k)}$-coalgebras.

(i) A *passive relation* provides for each $k \in K$ a semilattice relation $\mathcal{R}_k$ from $\mathsf{Pass}(k)(A, \zeta)$ to $\mathsf{Pass}(k)(A', \zeta')$. It is a *bisimulation* when $x\,\mathcal{R}_k\,x'$ implies that for each $i \in \mathsf{Ar}(k)$ we have $\zeta(x@i)\ \bigoplus_{k\in K} \mathcal{R}_k\ \zeta(x'@i)$.

(ii) Elements $M \in A$ and $M' \in A'$ are *bisimilar* when $\zeta(M)\ \bigoplus_{k\in K} \mathcal{R}_k\ \zeta(M')$ for some bisimulation $\mathcal{R}$.

Proposition 3.12 *Let (A, ζ) and (A', ζ') be $\bigoplus_{k\in K} \prod_{i\in \mathsf{Ar}(k)}$-coalgebras. Elements $M \in A$ and $M \in A'$ are trace equivalent iff they are bisimilar.*

Proof. In Appendix. ◻

4 Non-total systems

So far we have only considered total systems; but the more general case, where $\zeta(M)$ can be empty, is also of interest. For the language, this means we add a command `die` that serves as a unit for `or`. We summarize the changes required.

Definability of strategies becomes an easier problem: a strategy is

- definable by a finitely nondeterministic well-founded process iff it is anti-König
- definable by a countably nondeterministic well-founded process iff it is countably nondeterministic
- always definable by a well-founded process
- definable by a finitely nondeterministic process if it is finitely nondeterministic
- definable by a countably nondeterministic process iff it is countably nondeterministic
- always definable by a process.

A sound and complete theory of trace equivalence is given by the equations of a bounded semilattice (semilattice with least element) and commutativity of each operation with `or`. Note that we do not include the commutativity

$$\mathsf{in}_k(\mathtt{die})_{i\in\mathsf{Ar}(k)} = \mathtt{die}$$

as this is unsound for trace equivalence. Because of this exclusion, the set of anti-König strategies does not form an initial S-algebra on $\mathbf{Semilatt}^{\perp}$ (the category of bounded semilattices). However, it does form an initial algebra for the functor $F \bigoplus_{k\in K} U \prod_{i\in\mathsf{Ar}(k)}$ on $\mathbf{Semilatt}^{\perp}$, writing $U\colon \mathbf{Semilatt}^{\perp} \longrightarrow \mathbf{Semilatt}$ for inclusion and F for its right adjoint. (Note that we take a coproduct in $\mathbf{Semilatt}$, not $\mathbf{Semilatt}^{\perp}$.) That is because an algebra structure $F \bigoplus_{k \in K} U \prod_{i \in \mathsf{Ar}(k)} A \longrightarrow A$ corresponds to a family of (mere) semilattice homomorphisms $(f_k \colon U \prod_{i \in \mathsf{Ar}(k)} A \longrightarrow UA)_{k \in K}$, while an algebra morphism must preserve the least element as well as everything else, so the category of algebras for this functor is the category of models for our theory. The countably nondeterministic and unconstrained cases are similar.

Moreover, the set of finitely nondeterministic strategies forms a final coalgebra for this functor, by a similar argument to the one above, and all the other results of Section 3.2 hold correspondingly.

5 Conclusions

We have considered various notions of nondeterministic strategy in the sense of finite traces. They might at first sight appear *ad hoc*, but we have learnt that in each case they form an initial algebra or final coalgebra for a suitable functor.

We have chosen to work with coalgebras for an endofunctor $\mathcal{P}_{\mathsf{f}}^{+} \sum_{k \in K} \prod_{i \in \mathsf{Ar}(k)}$, because our language is then an example. But our narrative and proofs would also work for coalgebras for $\prod_{j \in j} \mathcal{P}_{\mathsf{f}}^{+} \sum_{k \in P(j)}$, or for many-sorted endofunctors [16].

Further work includes

- replacing nondeterministic by (finite or countable) probabilistic choice
- considering nondeterministic strategies in the sense of infinite traces [15].

A Appendix

Proposition 1.6: the least congruence $\equiv$ on commands satisfying

$$\begin{aligned} M \texttt{ or } N &\equiv N \texttt{ or } M \\ (M \texttt{ or } N) \texttt{ or } P &\equiv M \texttt{ or } (N \texttt{ or } P) \\ M \texttt{ or } M &\equiv M \end{aligned}$$

is bisimilarity.

Proof. We define operations on $\equiv$-classes:

$$\begin{aligned} \texttt{In}\, k(A_i)_{i \in \mathsf{Ar}(k)} &\stackrel{\text{def}}{=} [\texttt{In}\, k(M_i)_{i \in \mathsf{Ar}(k)}]_{\equiv} && \text{where } A_i = [M_i]_{\equiv} \\ A \texttt{ or } B &\stackrel{\text{def}}{=} [M \texttt{ or } N]_{\equiv} && \text{where } A = [M]_{\equiv} \text{ and } B = [N]_{\equiv} \\ \texttt{choose}\, \{A_0, \ldots, A_n\} &\stackrel{\text{def}}{=} A_0 \texttt{ or } \cdots \texttt{ or } A_n \end{aligned}$$

We inductively define the set of *bisimulation normal forms* as follows:

$$A \ ::= \quad \texttt{choose}\, \{\texttt{In}\, k_0(A_{0,i})_{i \in \mathsf{Ar}(k_0)}, \ldots, \texttt{In}\, k_n(A_{n,i})_{i \in \mathsf{Ar}(k_n)}\}$$

The bisimulation normal form of a command M is given by

$$\mathsf{NF}(M) \stackrel{\text{def}}{=} \texttt{choose}_{(k_0, x_0) \in \zeta M} \texttt{In}\, k_0(i_0 \mapsto \texttt{choose}_{(k_1, x_1) \in \zeta(x_0 @ i_0)} \texttt{In}\, k_1(i_1 \mapsto \cdots))$$

which is well-founded by Proposition 1.4(ii). Concisely:

$$\mathsf{NF}(M) \stackrel{\text{def}}{=} \texttt{choose}_{(k,x) \in \zeta M} \texttt{In}\, k(\mathsf{NF}(x @ i))_{i \in \mathsf{Ar}(k)} \tag{A.1}$$

For all M, we have $\mathsf{NF}(M) = [M]_{\equiv}$ by induction on M. If $M \sim N$ then $\mathsf{NF}(M) = \mathsf{NF}(N)$ by induction on $\mathsf{NF}(M)$ using (A.1). Finally, if $M \sim N$ then $[M]_{\equiv} = \mathsf{NF}(M) = \mathsf{NF}(N) = [N]_{\equiv}$. ◻

Proposition 1.16: for a nondeterministic strategy σ, the following are equivalent.

- $\sigma = \mathsf{Traces}\, M$ for some command M.
- σ is total and anti-König.

Proof. of ($\Leftarrow$). We inductively define the set of *trace normal forms.*

$$E \ ::= \quad \mathtt{choose}_{k \in L} \, \mathtt{In} \, k(E_{k,i})_{i \in \mathsf{Ar}(k)} \quad (L \in \mathcal{P}_{\mathsf{f}}^{+} K) \tag{A.2}$$

This grammar is more restrictive than the one for bisimulation normal forms: it does not allow two distinct choices to output k. We show that $M \mapsto \mathsf{Traces}\, M$ is a bijection from trace normal forms to total anti-König strategies. For a total anti-König strategy σ, its trace normal form $\mathsf{TNF}(\sigma)$ is given by

$$\mathsf{TNF}(\sigma) \stackrel{\text{def}}{=} \mathtt{choose}_{k_0 \in \mathsf{Resp}(\sigma,\varepsilon)} \mathtt{In}\, k_0(i_0 \mapsto \mathtt{choose}_{k_1 \in \mathsf{Resp}(\sigma, k_0 i_0)} \mathtt{In}\, k_1(i_1 \mapsto \ldots$$

which is well-founded by the anti-König property. Concisely:

$$\mathsf{TNF}(\sigma) \ = \ \mathtt{choose}_{k \in \mathsf{Init}\, \sigma} \mathtt{In}\, k \mathsf{TNF}(\sigma/ki)_{i \in \mathsf{Ar}(k)}$$

If σ is a total König strategy, then $\mathsf{TNF}(\sigma)$ is the unique trace normal form E such that $\mathsf{Traces}\, E = \sigma$. (To prove this, we show $\mathsf{Traces}\, \mathsf{TNF}(\sigma) = \sigma$ by induction on $\mathsf{TNF}(\sigma)$, and $\mathsf{TNF}(\mathsf{Traces}\, E) = E$ by induction on E.) The result follows. ◻

Theorem 1.18: for commands M and N, we have $M \equiv_{\mathsf{c}} N$ iff $\mathsf{Traces}\, M = \mathsf{Traces}\, N$.

Proof. ($\Rightarrow$) is evident. For ($\Leftarrow$), we first prove that, for a trace normal form E and $\equiv$-class A, if $\mathsf{Traces}\, E = \mathsf{Traces}\, A$ then $E =_{\mathsf{c}} A$. We proceed by induction on E. We have

$$\begin{aligned} E &= \mathtt{choose}_{k \in L} \mathtt{In}\, k(E_{k,i})_{i \in \mathsf{Ar}(k)} \\ A &= \mathtt{choose}\ \{\mathtt{In}\, k_0(A_{0,i})_{i \in \mathsf{Ar}(k_0)}, \ldots, \mathtt{In}\, k_n(A_{n,i})_{i \in \mathsf{Ar}(k_n)}\} \end{aligned}$$

Since $\mathsf{Traces}\, E = \mathsf{Traces}\, A$, we have $L = \{k_0, \ldots, k_n\}$, and for $k \in L$ and $i \in \mathsf{Ar}(k)$ we have

$$\begin{aligned} \mathsf{Traces}\, E_{k,i} &= \mathsf{Traces}\, \mathtt{choose}_{j \in Q_k} A_{j,i} \text{ where } Q_k \stackrel{\text{def}}{=} \{j \in [0\,..\,n] \mid k_j = k\} \\ \text{and so} \quad E_{k,i} &=_{\mathsf{c}} \mathtt{choose}_{j \in Q_k} A_{j,i} \end{aligned}$$

Now we reason

$$\begin{aligned} A &= \mathtt{choose}_{k \in L} \mathtt{choose}_{j \in Q_k} \mathtt{In}\, k(A_{j,i})_{i \in \mathsf{Ar}(k)} \\ &=_{\mathsf{c}} \mathtt{choose}_{k \in L} \mathtt{In}\, k(\mathtt{choose}_{j \in Q_k} A_{j,i})_{i \in \mathsf{Ar}(k)} \quad \text{by commutativity} \\ &=_{\mathsf{c}} \mathtt{choose}_{k \in L} \mathtt{In}\, k(E_{k,i})_{i \in \mathsf{Ar}(k)} \\ &= E \end{aligned}$$

Given commands M and N such that $\mathsf{Traces}\, M = \mathsf{Traces}\, N = \sigma$, put $E = \mathsf{TNF}(\sigma)$. Then $\mathsf{Traces}\, M = \mathsf{Traces}\, E = \mathsf{Traces}\, N$ gives $[M]_{\equiv} =_{\mathsf{c}} E =_{\mathsf{c}} [N]_{\equiv}$. ◻

Proposition 2.3: for any strategy σ, the following are equivalent.

- $\sigma = \mathsf{Traces}\, M$ for some command M.
- σ is well-foundedly total and countably nondeterministic.

Proof. ($\Rightarrow$) is explained in the text. For ($\Leftarrow$), we proceed as follows. First choose, for each $s \in \sigma^{\mathsf{AE}}$, a well-founded tree τ such that $t \in \tau$ implies $st \in \sigma$, and write $T(s)$ for the corresponding deterministic command. For $n \in \mathbb{N}$, form a command M_n that, for the first n cycles, follows the same pattern as trace normal form (Definition A.2), and then behaves deterministically:

$$\begin{aligned} M_n \stackrel{\text{def}}{=} \ & \mathtt{choose}_{k_0 \in \mathsf{Resp}(\sigma,\varepsilon)} \mathtt{In}\, k_0(i_0 \mapsto \\ & \quad \ldots \\ & \quad \mathtt{choose}_{k_{n-1} \in \mathsf{Resp}(\sigma, k_0 i_0 \ldots i_{n-2})} \mathtt{In}\, k_{n-1}(i_{n-1} \mapsto T(k_0 i_0 \ldots k_{n-1} i_{n-1}) \ldots)) \end{aligned}$$

Then $\mathsf{Traces}\, M_n \subseteq \sigma$, and for every $s \in \sigma$ with n outputs, $s \in \mathsf{Traces}\, M_n$. Therefore $\mathsf{Traces}\, \mathtt{choose}\ (M_n)_{n \in \mathbb{N}} = \sigma$. ◻

To prove Theorem 2.5, we cannot bring a command to trace normal form, but the following lemma shows that that we can at least bring it to a "top-level" version of trace normal form.

Lemma A.1 *For any* $\equiv$*-class* B*, we have* $B =_{\mathsf{c}} \mathtt{choose}_{k \in L} \mathtt{In}\, k(B_{k,i})_{i \in \mathsf{Ar}(k)}$ *for some* $L \in \mathcal{P}^{+}_{\mathsf{c}} K$ *and collection of* $\equiv$*-classes* $(B_{k,i})_{k \in L, i \in \mathsf{Ar}(k)}$.

Proof. We put $B = [M]_{\equiv}$ and proceed by induction on M. The case $M = \mathtt{In}\, k(M_i)_{i \in \mathsf{Ar}(k)}$ is trivial. Suppose $M = \mathtt{choose}\ (M_n)_{n \in \mathbb{N}}$ and $[M_n]_{\equiv} = \mathtt{choose}_{k \in L_n} \mathtt{In}\, k(B_{n,k,i})_{i \in \mathsf{Ar}(k)}$. Then

$$\begin{aligned} B &= \mathtt{choose}_{n \in \mathbb{N}, k \in L_n} \mathtt{In}\, k(B_{n,k,i})_{i \in \mathsf{Ar}(k)} \\ &= \mathtt{choose}_{k \in \bigcup_{n \in \mathbb{N}} L_n} \mathtt{choose}_{n \in Q_k} \mathtt{In}\, k(B_{n,k,i})_{i \in \mathsf{Ar}(k)} \quad \text{where } Q_k \stackrel{\text{def}}{=} \{n \in \mathbb{N} \mid k \in L_n\} \\ &= \mathtt{choose}_{k \in \bigcup_{n \in \mathbb{N}} L_n} \mathtt{In}\, k(\mathtt{choose}_{n \in Q_k} B_{n,k,i})_{i \in \mathsf{Ar}(k)} \end{aligned}$$

The case $M = M_0\ \mathtt{or}\ M_1$ is similar. □

Theorem 2.5: for commands M and N, we have $M \equiv_{\mathsf{c}} N$ iff $\mathsf{Traces}\, M = \mathsf{Traces}\, N$.

Proof. ($\Rightarrow$) is evident. For ($\Leftarrow$), we prove that, for $\equiv$-classes A and B, if $\mathsf{Traces}\, A \subseteq \mathsf{Traces}\, B$ then $A \leqslant_{\mathsf{c}} B$. (Here $\leqslant_{\mathsf{c}}$ is the partial order arising from $=_{\mathsf{c}}$ and $\mathtt{or}$.) We put $A = [M]_{\equiv}$ and proceed by induction on M. If $M = \mathtt{choose}\ (M_n)_{n \in \mathbb{N}}$ or $M = M_0\ \mathtt{or}\ M_1$ we apply the supremum property. If $M = \mathtt{In}\, k(M_i)_{i \in \mathsf{Ar}(k)}$, we apply Lemma A.1 to B:

$$\mathsf{Traces}\, \mathtt{In}\, k(M_i)_{i \in \mathsf{Ar}(k)} \subseteq \mathsf{Traces}\, \mathtt{choose}_{k \in L} \mathtt{In}\, k(B_{k,i})_{i \in \mathsf{Ar}(k)}$$

Thus $k \in L$, and for all $i \in \mathsf{Ar}(k)$ we have

$$\begin{aligned} \mathsf{Traces}\, M_i &\subseteq \mathsf{Traces}\, B_{k,i} \\ \text{so} \quad [M_i]_{\equiv} &\leqslant_{\mathsf{c}} B_{k,i} \end{aligned}$$

So $A \leqslant_{\mathsf{c}} \mathtt{In}\, k(B_{k,i})_{i \in \mathsf{Ar}(k)} \leqslant_{\mathsf{c}} B$. □

Theorem 2.7: the set of well-foundedly total strategies, ordered by inclusion, with $(\mathsf{In}\, k)_{k \in K}$, is an initial S-algebra in almost complete semilattices.

Proof. Let C be the set of well-founded total strategies. Put $\lambda \stackrel{\text{def}}{=} \aleph_0 \vee |K| \vee |C|$. Say that a λ*-semilattice* is a (join) semilattice where every family $(a_i)_{i<\lambda}$ has a supremum, and a *homomorphism* is a function that preserves these suprema. By the same proof as above, using a λ-ary nondeterministic choice operation, C forms an initial S-algebra in λ-semilattices. Let A be an S-algebra in almost complete semilattices, and $f : C \longrightarrow A$ the unique homomorphism of S-algebras in λ-semilattices. Any inhabited $R \subseteq C$ has cardinality $\leqslant \lambda$, so its supremum is preserved by f. Hence f is an almost complete semilattice homomorphism. □

Theorem 3.7: let $\zeta : X \longrightarrow \mathcal{P}_{\mathsf{f}} \sum_{k \in K} \prod_{i \in \mathsf{Ar}(k)} X$ be a transition system. Then $M \mapsto \mathsf{Traces}\, M$ is the unique map $h : (X, \zeta) \longrightarrow (\mathsf{Strat}^{+}_{\mathsf{f}}, \Psi)$.

Proof. A function $h : X \longrightarrow \mathsf{Strat}^{+}_{\mathsf{f}}$ is a map iff the following equations hold for all $M \in X$.

$$\{k \in K \mid k \in h(M)\} = \{k \in K \mid \exists x.\, M \stackrel{k}{\Rightarrow} x\} \tag{A.3}$$

$$\{s \mid k.i.s \in h(M)\} = \bigcup_{M \stackrel{k}{\Rightarrow} x} h(x@i) \tag{A.4}$$

Clearly (A.3)–(A.4) are satisfied by $h : M \mapsto \mathsf{Traces}\, M$. Conversely, suppose $h : X \longrightarrow \mathsf{Strat}^{+}_{\mathsf{f}}$ satisfies (A.3)–

(A.4). We show that $s \in h(M) \iff s \in \mathsf{Traces}\, M$ by induction on s.

$$\begin{array}{rlll} k \in h(M) & \iff M \overset{k}{\Rightarrow} x & \text{for some } x & \text{by (A.3)} \\ & \iff k \in \mathsf{Traces}\, M & & \end{array}$$

$$\begin{array}{rlll} k.i.s \in h(M) & \iff s \in h(x@i) & \text{for some } M \overset{k}{\Rightarrow} x & \text{by (A.4)} \\ & \iff s \in \mathsf{Traces}\, x@i & \text{for some } M \overset{k}{\Rightarrow} x & \text{by inductive hypothesis} \\ & \iff \text{for some } M \overset{k}{\Rightarrow} x & & \end{array}$$

□

Theorem 3.8: $(\mathsf{Strat}_{\mathsf{f}}^{+}, \Psi)$ is a final $\bigoplus_{k \in K} \prod_{i \in \mathsf{Ar}(k)}$-coalgebra. For a coalgebra (A, ζ), the unique coalgebra morphism $(A, \zeta) \longrightarrow (\mathsf{Strat}_{\mathsf{f}}^{+}, \Psi)$ is $M \mapsto \mathsf{Traces}\, M$.

Proof. Let (A, ζ) be a $\bigoplus_{k \in K} \prod_{i \in \mathsf{Ar}(k)}$-coalgebra. We say that a *k-passive state* is a family $x = (M_i)_{i \in \mathsf{Ar}(k)}$ of elements of A. For $M \in A$, a *trace* of M is a passive-ending play $s = k_0, i_0, \ldots, k_n$ such that

$$\begin{array}{rcll} \zeta(M) & = & (U_0, (y_{0,k})_{k \in U_0}) & k_0 \in U_0 \\ \zeta(y_{0,k_0}@i_0) & = & (U_1, (y_{1,k})_{k \in U_1}) & k_1 \in U_1 \\ & \cdots & & \\ \zeta(y_{n-1,k_{n-1}}@i_{n-1}) & = & (U_n, (y_{n,k})_{k \in U_n}) & k_n \in U_n \end{array}$$

The set of traces is written $\mathsf{Traces}\, M$.

We show that $M \mapsto \mathsf{Traces}\, M$ is a semilattice homomorphism. That is: for $R \in \mathcal{P}_{\mathsf{f}}^{+} A$, if a passive-ending play s is a trace of $\bigvee R$ then it is a trace of some $M \in R$. We proceed by induction on s. Suppose that s begins with k. For $M \in R$, we have

$$\begin{array}{rcl} \zeta(M) & = & (U_M, (y_{M,k})_{k \in U_M}) \\ \zeta(\bigvee R) & = & (U, (y_k)_{k \in U}) \end{array}$$

Since ζ is a semilattice homomorphism we have

$$U = \bigcup_{M \in R} U_M$$

So $k \in U$ iff there is $M \in R$ such that $k \in U_M$. If $s = (k)$ we are done; otherwise $s = k.i.s'$ and we may assume $k \in U$. We have

$$\begin{array}{rcl} y_k & = & \bigvee_{\substack{M \in R \\ k \in U_M}} y_{M,k} \\ \text{hence } y_k@i & = & \bigvee_{\substack{M \in R \\ k \in U_M}} (y_{M,k}@i) \end{array}$$

Now s is a trace of $\bigvee R$ iff s' is a trace of $y_k@i$. By the inductive hypothesis, this is equivalent to the existence of $M \in R$ such that s' is a trace of $y_{M,k}@i$ i.e. such that s is a trace of M.

We show that $M \mapsto \mathsf{Traces}\, M$ is the unique function $h\colon A \longrightarrow \mathsf{Strat}_{\mathsf{f}}^{+}$ such that

$$\begin{array}{ccc} A & \xrightarrow{\quad h \quad} & \mathsf{Strat}_{\mathsf{f}}^{+} \\ {\scriptstyle\zeta}\downarrow & & \downarrow{\scriptstyle\Psi} \\ \bigoplus_{k\in K}\prod_{i\in\mathsf{Ar}(k)} A & \xrightarrow[\bigoplus_{k\in K}\prod_{i\in\mathsf{Ar}(k)} h]{} & \bigoplus_{k\in K}\prod_{i\in\mathsf{Ar}(k)} \mathsf{Strat}_{\mathsf{f}}^{+} \end{array}$$

This diagram says that for any $M \in A$ such that $\zeta M = (U, (y_k)_{k\in U})$, we have

$$\begin{aligned} \mathsf{Init}\, h(M) &= U \\ h(M)/ki &= h(y_k@i) \qquad \text{for } k \in U \text{ and } i \in \mathsf{Ar}(k) \end{aligned}$$

Clearly $M \mapsto \mathsf{Traces}\, M$ satisfies these. For any h satisfying them, we show for a passive-ending play s that $s \in h(M)$ iff $s \in \mathsf{Traces}\, M$, by induction on s, separating the cases $s = (k)$ and $s = k.i.s'$. □

Proposition 3.12: let (A, ζ) and (A', ζ') be $\bigoplus_{k\in K}\prod_{i\in\mathsf{Ar}(k)}$-coalgebras. Elements $M \in A$ and $M \in A'$ are trace equivalent iff they are bisimilar.

Proof. Although we have defined bisimilarity in terms of a passive bisimulation, it is easily shown equivalent to the corresponding notion defined in terms of an active bisimulation, i.e. a semilattice relation $A \overset{\mathcal{R}}{\longrightarrow} A'$ such that if $M\,\mathcal{R}\,M'$ then $\zeta M \bigoplus_{k\in K}\prod_{i\in\mathsf{Ar}(k)} \zeta'(M')$.

The identity relation on any coalgebra is an (active) bisimulation, and the inverse image of a bisimulation along coalgebra maps is a bisimulation. Hence the inverse image $\mathcal{R}$ of the identity relation on $(\mathsf{Strat}_{\mathsf{f}}^{+}, \Psi)$ along the coalgebra maps $(A, \zeta) \longrightarrow (\mathsf{Strat}_{\mathsf{f}}^{+}, \Psi)$ and $(A', \zeta') \longrightarrow (\mathsf{Strat}_{\mathsf{f}}^{+}, \Psi)$ is a bisimulation. This is trace equivalence, so trace equivalence implies bisimilarity.

Conversely, given an active bisimulation $\mathcal{R}$, let $B \stackrel{\text{def}}{=} f\{(N, N') \in A \times A' \mid \zeta(N) \bigoplus_{k\in K} \mathcal{R}_k\, \zeta(N')\}$, ordered componentwise. Binary joins are given componentwise, since $\mathcal{R}$ is a semilattice relation. For $(N, N') \in B$ with $\zeta N = (U, (y_k)_{k\in k}$ and $\zeta N' = (U, (y'_k)_{k\in K}$ we put $\xi(N, N') \stackrel{\text{def}}{=} (U, (z_k)_{k\in U})$ where $z_k@i \stackrel{\text{def}}{=} (y_k@i, y'_k@i)$ for $i \in \mathsf{Ar}(k)$. Then ξ is a semilattice homomorphism because ζ and ζ' are. The projections from (B, ξ) to (A, ζ) and (A', ζ') are coalgebra morphisms so if $M\,\mathcal{R}\,M'$ then $\mathsf{Traces}\, M = \mathsf{Traces}\,(M, M') = \mathsf{Traces}\, M'$. □

References

[1] Nathan Bowler, Sergey Goncharov, Paul Blain Levy, and Lutz Schröder. Exploring the boundaries of monad tensorability on set. *Logical Methods in Computer Science*, 9(3), 2013.

[2] Yuxin Deng, Yuan Feng, and Ugo Dal Lago. On coinduction and quantum lambda calculi. In Luca Aceto and David de Frutos-Escrig, editors, *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015*, volume 42 of *LIPIcs*, pages 427–440. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[3] Christian Eisentraut, Holger Hermanns, and Lijun Zhang 0001. Concurrency and composition in a stochastic world. In Paul Gastin and François Laroussinie, editors, *CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings*, volume 6269 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2010.

[4] P. Freyd. Algebra valued functors in general and tensor products in particular. *Colloquium Mathematicae*, 14(1), 1966.

[5] S. Goncharov and L. Schröder. Powermonads and tensors of unranked effects. In *LICS*, 2011.

[6] R. Harmer. *Games and full abstraction for nondeterministic languages.* PhD thesis, Univ. of London, 1999.

[7] R Harmer and G McCusker. A fully abstract game semantics for finite nondeterminism. In *14th Symposium on Logic in Comp. Sci.* IEEE, 1999.

[8] I. Hasuo, B. Jacobs, and A. Sokolova. Generic trace semantics via coinduction. *Logical Methods in Computer Science*, 3(4), 2007.

[9] Matthew Hennessy. Exploring probabilistic bisimulations, part i. *Formal Asp. Comput*, 24(4-6):749–768, 2012.

[10] Holger Hermanns, Jan Krcál, and Jan Kretínský. Probabilistic bisimulation: Naturally on distributions. *CoRR*, abs/1404.5084, 2014.

[11] J. M. E. Hyland, G. D. Plotkin, and A. J. Power. Combining effects: sum and tensor. *Theoretical Computer Science*, 357, 2006.

[12] B. Jacobs, P. B. Levy, and J. Rot. Steps and traces. to appear, 14th IFIP WG 1.3 International Workshop on Coalgebraic Methods in Computer Science (CMCS), 2018.

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

[14] Bartek Klin and Jurriaan Rot. Coalgebraic trace semantics via forgetful logics. In Andrew M. Pitts, editor, *Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, volume 9034 of *Lecture Notes in Computer Science*, pages 151–166. Springer, 2015.

[15] P. B. Levy. Infinite trace equivalence. *Annals of Pure & Applied Logic*, 151(2–3), 2008.

[16] Paul Blain Levy and Sam Staton. Transition systems over games. In Thomas A. Henzinger and Dale Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 64:1–64:10. ACM, 2014.

[17] E Moggi. Notions of computation and monads. *Information and Computation*, 93, 1991.

[18] G. Plotkin and J. Power. Notions of computation determine monads. In *Proceedings, Foundations of Software Science and Computation Structures, 2002*, volume 2303 of *LNCS*, pages 342–356. Springer, 2002.

[19] G. D. Plotkin and A. J. Power. Adequacy for algebraic effects. *LNCS*, 2030, 2001.

[20] Davide Sangiorgi and Valeria Vignudelli. Environmental bisimulations for probabilistic higher-order languages. In Rastislav Bodík and Rupak Majumdar, editors, *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016*, pages 595–607. ACM, 2016.