The Theory of Traces for Nondeterminism and Probability

Ana Sokolova

Coalgebra Day @ NII, Tokyo 28.10.19
It’s all about leaving a trace...
Joint work with

Ichiro Hasuo
Radboud University

Bart Jacobs

Alexandra Silva

Harald Woracek

Filippo Bonchi

Valeria Vignudelli

Ana Sokolova

NII Tokyo 28.10.19
I will talk about:

1. The absolute basics of coalgebra
2. Trace semantics via determinisation
3. …enabled by algebraic structure
I will talk about:

1. The absolute basics of coalgebra
2. Trace semantics via determinisation
3. ...enabled by algebraic structure
I will talk about:

1. The absolute basics of coalgebra

2. Trace semantics via determinisation

3. …enabled by algebraic structure

Mathematical framework based on category theory for state-based systems semantics for nondeterministic/probabilistic systems
I will talk about:

1. The absolute basics of coalgebra
2. Trace semantics via determinisation
3. …enabled by algebraic structure

Mathematical framework based on category theory for state-based systems semantics

for nondeterministic/probabilistic systems

systems with algebraic effects
Coalgebras

Uniform framework for dynamic transition systems, based on category theory.
Coalgebras

Uniform framework for dynamic transition systems, based on category theory.

\[ X \xrightarrow{c} FX \]
Coalgebras

Uniform framework for dynamic transition systems, based on category theory.

\[ X \xrightarrow{c} FX \]

states
Coalgebras

Uniform framework for dynamic transition systems, based on category theory.

\[ X \xrightarrow{c} FX \]
Coalgebras

Uniform framework for dynamic transition systems, based on category theory.

\[ X \xrightarrow{c} FX \]

states \hspace{2cm} type \hspace{2cm} object in the base category \( C \)
Coalgebras

Uniform framework for dynamic transition systems, based on category theory.

\[ X \xrightarrow{c} FX \]

- states
- type
- object in the base category \( \mathbf{C} \)
- functor on the base category \( \mathbf{C} \)
Coalgebras

Uniform framework for dynamic transition systems, based on category theory.

\[ X \xrightarrow{c} FX \]

states

object in the base category \( C \)

functor on the base category \( C \)

type

form a category too
Coalgebras

Uniform framework for dynamic transition systems, based on category theory.

\[ X \xrightarrow{c} FX \]

- States
- Type

Object in the base category \( \mathcal{C} \)

Functor on the base category \( \mathcal{C} \)

Generic notion of behavioural equivalence (bisimilarity)

\( \approx \)

Form a category too
Examples

NFA

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

\[ a, b \]

\[ x \downarrow_0 \quad b \rightarrow \quad y \downarrow_1 \]
We introduce now the main approach and show that we can recover known trace semantics from the literature. We use the generalised determinisation [30], providing e.g. an abstract tool, since it provides a finite syntax for describing finite traces and trace semantics [22] for nondeterministic probabilistic systems that unifies the treatment of language equivalence for automata. We use the local and the global view. We start by taking the global-view approach founded on algebras and coalgebras and inspired via generalised determinisation.

In this paper, we propose a theory of trace semantics for such processes can be recovered by instantiating a coalgebraic construction known to each state. Here recall nondeterministic automata (NDA) and Rabin 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 [2], [3], [4], [5], [6], [7], and unknown environment, implementation freedom, or concurrency. At the same time, the interplay of nondeterminism enables modelling of incomplete information, determinism.

Abstract

Figure 1 shows a nondeterministic probabilistic system (NPLTS) that we use as a running example. The type of choice, modelled abstractly by a monad $\mathbb{P}X$ is often linked to a concrete algebraic theory, the algebraic theory of convex algebras. Once we have such an algebraic presentation, we have a determinised tool, since it provides a finite syntax for describing finite pieces of our puzzle, and show how everything combines. In order to illustrate our approach, it is convenient to compare the resulting semantics to known definitions of trace equivalences appearing in the literature. Most of our results are based on such auxilary notions of resolutions and [26] are based on such auxilary notions of resolutions and equivalences.

Examples

NFA

$X \rightarrow 2 \times (\mathcal{P}X)^A$

$\xymatrix{a, b \ar@/^/[r] & x \downarrow_0 \ar[r] & b \ar[r] & y \downarrow_1}$

Rabin PA

$X \rightarrow [0,1] \times (\mathcal{P}_{\leq 1}X)^A$

$\xymatrix{x \downarrow_0 \ar[r]^a \ar[r]^b \ar@{.>}[r] & y \downarrow_1}$

$\xymatrix{\frac{1}{2} \ar@/^/[r] & a, b \ar[r]^a \ar[r]^b \ar@{.>}[r] & y \downarrow_1}$
Examples

NFA

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

- a, b
- \( x \downarrow_0 \rightarrow b \rightarrow y \downarrow_1 \)

Simple NPA

\[ X \rightarrow ? \times (\mathcal{P}D\mathcal{X})^A \]

Rabin PA

\[ X \rightarrow [0,1] \times (\mathcal{P}\leq_1 X)^A \]

- \( x \downarrow_0 \rightarrow \frac{1}{2} \rightarrow a, b \rightarrow \frac{1}{2} \rightarrow y \downarrow_1 \)
- \( x \rightarrow \frac{1}{2} \rightarrow a, b \rightarrow \frac{1}{2} \rightarrow y \)
Examples

NFA

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

\[ a, b \]

\[ x \downarrow_0 \rightarrow b \rightarrow y \downarrow_1 \]

Simple NPA

\[ X \rightarrow \? \times (\mathcal{P}OX)^A \]

Rabin PA

\[ X \rightarrow [0,1] \times (\mathcal{D}_{\leq 1}X)^A \]

\[ x \downarrow_0 \overset{a,b}{\rightarrow} \overset{\frac{1}{2}}{\cdots \cdots} \rightarrow y \downarrow_1 \overset{a,b}{\rightarrow} \]

systems with nondeterminism and probability

This is the approach of trace semantics via a determinisation. For example, in the determinised PA we have, since there is a close relation between theauty of a state and the probability of its appearance, we can recover known trace semantics by using determinisation.

Abstract

The Theory of Traces for Systems with Nondeterminism and Probability

We develop a coalgebraic theory of traces for systems combining nondeterministic and probabilistic choices. This framework unifies the different kinds of systems and automata, via Kleisli traces [29].

Probability is needed to quantitatively model uncertainty and belief, whereas nondeterminism is usually used to model the behavior of the environment, in which the system that we study is not in control.

The theory of coalgebra [27], [28] provides an abstract framework for studying systems.

The theory of convex semilattices [20], [21], [32], namely, the monad \( (\mathcal{P}O \times [0,1])^A \), is generated by a singleton set. We prove their concrete characterisation and recover them by instantiating a coalgebraic construction known as the generalised powerset construction. We characterise and compare the resulting semantics to known definitions of trace and testing semantics (with tests being finite presentations via algebraic theories).

In our opinion, drastically clarifies and simplifies the trace theory of systems with nondeterminism and probability. Putting it to good use is part of our contribution which, together in the theory of traces for NPLTS.

In Section 4 we present the language and semantics of systems combining nondeterminism and probability. We then recall, in Section 5, the generalised determinisation technique which gives rise to the determinisation monad \( (\mathcal{P}O \times [0,1])^A \).

Section 3 (Theorem 4) and combine it with termination time equivalences for NPLTS, namely bisimilarity and \( \mu \)-simulation. The equivalence is implied by the standard branching-time equivalences appearing in the literature in the fact that they are reactive rather than fully synchronous.

The greatest probability with which a state passes a given test; the output value after executing a trace by following the trace is the greatest probability with which a state passes a given test. The output value after executing a trace by following the trace is the greatest probability with which a state passes a given test.

This is the approach of trace semantics via a determinisation. For example, in the determinised PA we have, since there is a close relation between theauty of a state and the probability of its appearance, we can recover known trace semantics by using determinisation.
In general
In general

Automata

\[ X \rightarrow O \times (MX)^A \]
In general

Automata

\[ X \rightarrow O \times (MX)^A \]

with observations in O
In general

Automata

$X \rightarrow O \times (MX)^A$

with observations in $O$

and M-effects
In general

Automata

\[ X \rightarrow O \times (MX)^A \]

with observations in O

and M-effects

for a monad M
In general

Automata

\[ X \rightarrow \text{O} \times (\text{MX})^A \]

and M-effects

for a monad \( M \)

\[ \mu: MM \Rightarrow M \]
\[ \eta: Id \Rightarrow M \]

with observations in \( \text{O} \)
In general

Automata

\[ X \rightarrow O \times (MX)^A \]

with observations in \( O \)

and M-effects

for a monad \( M \)

\[ \mu : MM \Rightarrow M \]
\[ \eta : Id \Rightarrow M \]

providing algebraic effects
In general

Automata

\[ X \rightarrow O \times (MX)^A \]

with observations in \( O \)

and M-effects

for a monad \( M \)

we write \( x \downarrow O, x \xrightarrow{a} t_x \)

\[ \mu: MM \Rightarrow M \]
\[ \eta: \text{Id} \Rightarrow M \]

providing algebraic effects
In our examples

NFA
\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

Rabin PA
\[ X \rightarrow [0,1] \times (\mathcal{P}_{\leq 1}X)^A \]

Simple PA
\[ X \rightarrow ? \times (\mathcal{P}D^A) \]
In our examples

NFA

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

M = \mathcal{P}

for nondeterminism

Rabin PA

\[ X \rightarrow [0, 1] \times (\mathcal{D}_{\leq 1}X)^A \]

Simple PA

\[ X \rightarrow ? \times (\mathcal{PO}X)^A \]
In our examples

NFA

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

M = \mathcal{P} for nondeterminism

Powerset, subsets

Rabin PA

\[ X \rightarrow [0,1] \times (\mathcal{P}_{\leq 1}X)^A \]

Simple PA

\[ X \rightarrow ? \times (\mathcal{P}DX)^A \]
In our examples

NFA

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

M = \mathcal{P}

for nondeterminism

Powerset, subsets

Rabin PA

\[ X \rightarrow [0,1] \times (\mathcal{D}_{\leq 1} X)^A \]

M = \mathcal{D}_{\leq 1}

for probability

Simple PA

\[ X \rightarrow ? \times (\mathcal{P}D X)^A \]
In our examples

NFA

\[ X \rightarrow 2 \times (\mathcal{P}X)^\mathcal{A} \]

M = \mathcal{P} for nondeterminism

Powerset, subsets

Rabin PA

\[ X \rightarrow [0,1] \times (\mathcal{D}_{\leq 1}X)^\mathcal{A} \]

M = \mathcal{D}_{\leq 1} for probability

Subdistributions

Simple PA

\[ X \rightarrow ? \times (\mathcal{PDX})^\mathcal{A} \]
In our examples

- **NFA**
  \[ X \rightarrow 2 \times (\mathcal{P}X)^A \]
  \( M = \mathcal{P} \)
  for nondeterminism
  Powerset, subsets

- **Rabin PA**
  \[ X \rightarrow [0,1] \times (\mathcal{D}_{\leq 1}X)^A \]
  \( M = \mathcal{D}_{\leq 1} \)
  for probability
  Subdistributions

- **Simple PA**
  \[ X \rightarrow ? \times (\mathcal{PD}X)^A \]
  \( M = \mathcal{PD} ??? \)
  for nondeterminism and probability
In our examples

**NFA**

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

\[ M = \mathcal{P} \]

for nondeterminism

**Powerset, subsets**

**Rabin PA**

\[ X \rightarrow [0, 1] \times (\mathcal{D}_{\leq 1}X)^A \]

\[ M = \mathcal{D}_{\leq 1} \]

for probability

**Subdistributions**

**Simple PA**

\[ X \rightarrow ? \times (\mathcal{C}X)^A \]

\[ M = \mathcal{C} \]

for nondeterminism and probability!
In our examples

NFA

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

\[ M = \mathcal{P} \]

for nondeterminism

Powerset, subsets

Rabin PA

\[ X \rightarrow [0,1] \times (\mathcal{D}_{\leq 1}X)^A \]

\[ M = \mathcal{D}_{\leq 1} \]

for probability

Subdistributions

Simple PA

\[ X \rightarrow ? \times (\mathcal{C}X)^A \]

\[ M = \mathcal{C} \]

for nondeterminism and probability!

Nonempty f.g. convex subsets of distributions
Trace Semantics
Trace Semantics

language semantics
Trace Semantics

NFA = LTS + termination

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

language semantics

[Diagram showing a transition labeled with symbols a, b, and transitions labeled x \Downarrow_0 \rightarrow b \rightarrow y \Downarrow_1]
Trace Semantics

NFA = LTS + termination

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

\[
\begin{array}{c}
\text{a, b} \\
\xrightarrow{x \downarrow_0} \xrightarrow{b} \xrightarrow{y \downarrow_1}
\end{array}
\]

\[ \text{tr}: X \rightarrow 2^A^* \]
Trace Semantics

NFA = LTS + termination

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

\[ \begin{array}{ccc}
  a, b \\
  x \downarrow_0 \rightarrow b \rightarrow y \downarrow_1 \\
\end{array} \]

\( \text{tr} : X \rightarrow 2^A^* \)

\[ \text{tr}(x) = (a \cup b)^*b = \{ w \in \{a, b\}^* \mid w \text{ ends with a } b \} \]
Trace Semantics

Rabin PA = RPTS + termination

\[ X \rightarrow [0,1] \times (\mathcal{P}_{\leq 1} X)^A \]

\[ \begin{align*}
\text{tr}(x) &= \left( a \mapsto \frac{1}{2}, \; aa \mapsto \frac{3}{4}, \ldots \right) \\
\text{tr}: X &\rightarrow [0,1]^A^* 
\end{align*} \]
Trace Semantics

Rabin PA = RPTS + termination

\[ X \rightarrow [0,1] \times (\mathcal{P}_{\leq 1} X)^A \]

\[ \begin{array}{c}
\text{tr}(x) = \left( a \mapsto \frac{1}{2}, aa \mapsto \frac{3}{4}, \ldots \right) \\
\text{tr}: X \rightarrow [0,1]^A^* \\
\end{array} \]
Trace Semantics

Simple NPA

\[ X \rightarrow ? \times (\mathcal{POX})^A \]

\[ \text{tr}(x) = ??? \]

\[ \text{tr}: X \rightarrow ?^A \]

We develop a theory of traces for NPLTS using such an approach. For this purpose we take the monad for nondeterminism and probability as coalgebras and computational effects as monads.
Trace Semantics

Simple NPA

\[ X \rightarrow ? x (\mathcal{PD}X)^A \]

\[ \text{tr}(x) = ??? \]

\[ \text{tr}: X \rightarrow ?^A \]

Nondet. probabilistic language semantics?
Trace Semantics

Simple NPA

\[ X \rightarrow \ ? x (\mathcal{PD}X)^A \]

\[ \text{tr}(x) = ??? \]

\[ \text{tr}: X \rightarrow ?^A \]

-existing definitions are “local”
given in terms of schedulers

nondet. probabilistic language semantics?

The induced equivalence can be proved coinductively by means of proof-techniques known as +

coincides with the local one, namely that our three semantics

The equivalence is implied by the standard branching-

The equivalence coincides with our may trace equivalence

When regarding an LTS and RPLTS as
time equivalences for NPLTS, namely bisimilarity and

distinguishing between deterministic and probabilistic. We observe that however this difference does

ability with which the trace can be performed, with respect

may-trace semantics assigns to each trace the greatest prob-

coincides with the local one, namely that our three semantics

assigns the smallest one. It is important to remark here that

probability over nondeterminism. Having

identifying what are the

A

on this journey. Having the presentation enables us to clearly

theory of systems with nondeterminism and probability.

in our opinion, drastically clarifies and simplifies the trace

putting it to good use is part of our contribution which,

not explicitly proven, in the community — proving it and

for the NPLTS from Figure 1.

We develop a theory of traces for NPLTS using such

We call the resulting three semantics

executing the trace

of observations: a semilattice on

subsets of distributions, and provide all necessary and con-

theory of pointed convex semilattices

identify what are the

on this journey. Having the presentation enables us to clearly

theory of systems with nondeterminism and probability.

in our opinion, drastically clarifies and simplifies the trace

putting it to good use is part of our contribution which,

not explicitly proven, in the community — proving it and

for the NPLTS from Figure 1.

We develop a theory of traces for NPLTS using such

We call the resulting three semantics

executing the trace

of observations: a semilattice on

subsets of distributions, and provide all necessary and con-

theory of pointed convex semilattices

identify what are the

on this journey. Having the presentation enables us to clearly

theory of systems with nondeterminism and probability.

in our opinion, drastically clarifies and simplifies the trace

putting it to good use is part of our contribution which,

not explicitly proven, in the community — proving it and

for the NPLTS from Figure 1.

We develop a theory of traces for NPLTS using such

We call the resulting three semantics

executing the trace

of observations: a semilattice on

subsets of distributions, and provide all necessary and con-

theory of pointed convex semilattices

identify what are the

on this journey. Having the presentation enables us to clearly

theory of systems with nondeterminism and probability.

in our opinion, drastically clarifies and simplifies the trace

putting it to good use is part of our contribution which,

not explicitly proven, in the community — proving it and

for the NPLTS from Figure 1.

We develop a theory of traces for NPLTS using such

We call the resulting three semantics

executing the trace

of observations: a semilattice on

subsets of distributions, and provide all necessary and con-

theory of pointed convex semilattices

identify what are the

on this journey. Having the presentation enables us to clearly

theory of systems with nondeterminism and probability.

in our opinion, drastically clarifies and simplifies the trace

putting it to good use is part of our contribution which,

not explicitly proven, in the community — proving it and

for the NPLTS from Figure 1.

We develop a theory of traces for NPLTS using such

We call the resulting three semantics

executing the trace

of observations: a semilattice on

subsets of distributions, and provide all necessary and con-

theory of pointed convex semilattices

identify what are the

on this journey. Having the presentation enables us to clearly

theory of systems with nondeterminism and probability.

in our opinion, drastically clarifies and simplifies the trace

putting it to good use is part of our contribution which,

not explicitly proven, in the community — proving it and

for the NPLTS from Figure 1.

We develop a theory of traces for NPLTS using such

We call the resulting three semantics

executing the trace

of observations: a semilattice on

subsets of distributions, and provide all necessary and con-

theory of pointed convex semilattices

identify what are the

on this journey. Having the presentation enables us to clearly

theory of systems with nondeterminism and probability.

in our opinion, drastically clarifies and simplifies the trace

putting it to good use is part of our contribution which,

not explicitly proven, in the community — proving it and

for the NPLTS from Figure 1.

We develop a theory of traces for NPLTS using such

We call the resulting three semantics

executing the trace

of observations: a semilattice on

subsets of distributions, and provide all necessary and con-

theory of pointed convex semilattices

identify what are the

on this journey. Having the presentation enables us to clearly

theory of systems with nondeterminism and probability.

in our opinion, drastically clarifies and simplifies the trace

putting it to good use is part of our contribution which,

not explicitly proven, in the community — proving it and

for the NPLTS from Figure 1.

We develop a theory of traces for NPLTS using such

We call the resulting three semantics

executing the trace

of observations: a semilattice on

subsets of distributions, and provide all necessary and con-

theory of pointed convex semilattices

identify what are the

on this journey. Having the presentation enables us to clearly

theory of systems with nondeterminism and probability.

in our opinion, drastically clarifies and simplifies the trace

putting it to good use is part of our contribution which,

not explicitly proven, in the community — proving it and

for the NPLTS from Figure 1.

We develop a theory of traces for NPLTS using such

We call the resulting three semantics

executing the trace

of observations: a semilattice on

subsets of distributions, and provide all necessary and con-

theory of pointed convex semilattices

identify what are the

on this journey. Having the presentation enables us to clearly

theory of systems with nondeterminism and probability.

in our opinion, drastically clarifies and simplifies the trace

putting it to good use is part of our contribution which,

not explicitly proven, in the community — proving it and

for the NPLTS from Figure 1.

We develop a theory of traces for NPLTS using such

We call the resulting three semantics

executing the trace

of observations: a semilattice on

subsets of distributions, and provide all necessary and con-

theory of pointed convex semilattices

identify what are the

on this journey. Having the presentation enables us to clearly

theory of systems with nondeterminism and probability.

in our opinion, drastically clarifies and simplifies the trace

putting it to good use is part of our contribution which,

not explicitly proven, in the community — proving it and

for the NPLTS from Figure 1.
Trace semantics coalgebraically?

Two ideas:

1. unfold branching + transitions on words
2. trace = bisimilarity after determinisation
Trace semantics coalgebraically?

Two ideas:

1. unfold branching + transitions on words
2. trace = bisimilarity after determinisation
Trace semantics coalgebraically?

Two ideas:

1. unfold branching + transitions on words
2. trace = bisimilarity after determinisation

NFA / LTS
Trace semantics coalgebraically?

Two ideas:

1. unfold branching + transitions on words
2. trace = bisimilarity after determinisation
Trace semantics coalgebraically?

Two ideas:

1. unfold branching + transitions on words
2. trace = bisimilarity after determinisation

NFA / LTS
Trace semantics coalgebraically?

Two ideas:

1. unfold branching + transitions on words
2. trace = bisimilarity after determinisation
Trace semantics coalgebraically?

Two ideas:

1. unfold branching + transitions on words
2. trace = bisimilarity after determinisation
Trace semantics coalgebraically?

Two ideas:

(1) unfold branching + transitions on words

(2) trace = bisimilarity after determinisation
Trace semantics coalgebraically
Trace semantics coalgebraically

Two approaches:

(1) modelling in a Kleisli category

(2) modelling in an Eilenberg-Moore category
Trace semantics coalgebraically

Two approaches:

(1) modelling in a Kleisli category

(2) modelling in an Eilenberg-Moore category
Trace semantics coalgebraically

Two approaches:

(1) modelling in a Kleisli category

(2) modelling in an Eilenberg-Moore category

Hasuo, Jacobs, S. LMCS ’07
Trace semantics coalgebraically

Two approaches:

(1) modelling in a Kleisli category

(2) modelling in an Eilenberg-Moore category

algebras of a monad $M$
Trace semantics coalgebraically

Two approaches:

1. modelling in a Kleisli category
2. modelling in an Eilenberg-Moore category

algebras of a monad \( M \)

(1) and (2) are related

Hasuo, Jacobs, S. LMCS ’07
Silva, Bonchi, Bonsangue, Rutten FSTTCS’10
Trace semantics coalgebraically

Two approaches:

(1) modelling in a Kleisli category

(2) modelling in an Eilenberg-Moore category

(1) and (2) are related

Hasuo, Jacobs, S. LMCS ’07
Silva, Bonchi, Bonsangue, Rutten FSTTCS’10
Jacobs, Silva, S. JCSS’15
Traces via determinisation
Traces via determinisation

Automaton with M-effects

\[ X \rightarrow O \times (MX)^A \]
Traces via determinisation

Automaton with M-effects

\[ X \rightarrow O \times (MX)^A \]
Traces via determinisation

Automaton with M-effects

\[ X \rightarrow O \times (MX)^A \]

Determinisation

\[ MX \rightarrow O \times (MX)^A \]
Traces via determinisation

Automaton with M-effects

\[ X \rightarrow O \times (MX)^A \]

Determinisation

\[ MX \rightarrow O \times (MX)^A \]

trace = bisimilarity after determinisation
Traces via determinisation

Automaton with M-effects

\[ X \rightarrow O \times (MX)^A \]

Determinisation

\[ MX \rightarrow O \times (MX)^A \]

trace = bisimilarity after determinisation
Traces via determinisation

Automaton with M-effects

\[ X \rightarrow O \times (MX)^A \]

Determinisation

\[ MX \rightarrow O \times (MX)^A \]

trace = bisimilarity after determinisation

Algebras for M

ideally we have a presentation
Traces via determinisation

Automaton with M-effects

\[ X \rightarrow O \times (MX)^A \]

Determinisation

\[ MX \rightarrow O \times (MX)^A \]

\( O \) has to be an M-algebra!

trace = bisimilarity after determinisation

Algebras for M

ideally we have a presentation

NII Tokyo 28.10.19
Traces via determinisation

Automaton with M-effects

\[ X \rightarrow O \times (MX)^A \]

Determinisation

\[ MX \rightarrow O \times (MX)^A \]

- O has to be an M-algebra!
- MX carries the free M-algebra
- Algebras for M
- Ideally we have a presentation

trace = bisimilarity after determinisation

Ana Sokolova
Traces via determinisation

Automaton with M-effects

\[ X \rightarrow O \times (MX)^A \]

Determinisation

\[ MX \rightarrow O \times (MX)^A \]

- O has to be an M-algebra!
- Trace = bisimilarity after determinisation

MX carries the free M-algebra

Algebras for M

Eilenberg-Moore algebras

Ideally we have a presentation
Traces via determinisation

Automaton with M-effects

\[ X \rightarrow O \times (MX)^A \]

Determinisation

\[ MX \rightarrow O \times (MX)^A \]

- O has to be an M-algebra!
- \( MX \) carries the free M-algebra
- Algebras for M
- Eilenberg-Moore algebras
- ideally we have a presentation

trace = bisimilarity after determinisation

\[ \text{tr}: X \rightarrow O^{A^*} \]
Traces via determinisation

Automaton with M-effects

$X \rightarrow O \times (MX)^A$

O has to be an M-algebra!

Determinisation

$MX \rightarrow O \times (MX)^A$

$MX$ carries the free M-algebra

Algebras for M

Eilenberg-Moore algebras

ideally we have a presentation

trace = bisimilarity after determinisation

$\text{tr}: X \rightarrow O^{A^*}$

$\text{tr}(x)(a_1 a_2 \ldots a_n) = o \iff x \xrightarrow{a_1} t_1 \xrightarrow{a_2} t_2 \ldots t_{n-1} \xrightarrow{a_n} t_n \land t_n \downarrow o$
Eilenberg-Moore Algebras

abstractly

- objects

\[
\begin{array}{c}
MA \\
\|^{a} \\
A
\end{array}
\]

\[
A \xrightarrow{\eta} MA \\
\downarrow^{a} \\
A
\]

\[
\begin{array}{c}
MMA \\
\|^{\mu} \\
MA \\
\|^{a} \\
A
\end{array}
\]

\[
MA \xrightarrow{a} A
\]

- morphisms

\[
\begin{array}{c}
MA \\
\|^{a} \\
A
\end{array}
\xrightarrow{h}
\begin{array}{c}
MB \\
\|^{b} \\
B
\end{array}
\]

\[
\begin{array}{c}
MA \\
\|^{Mh} \\
MB \\
\|^{a} \\
A
\end{array}
\xrightarrow{a} \\
\begin{array}{c}
MB \\
\|^{b} \\
B
\end{array}
\]

\[
\begin{array}{c}
A \\
\|^{h} \\
B
\end{array}
\xrightarrow{h}
\begin{array}{c}
A \\
\|^{b} \\
B
\end{array}
\]
Traces via determinisation
Traces via determinisation

NFA

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]
Traces via determinisation

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

\[ \xrightarrow{a, b} \]

\[ x \Downarrow_0 \xrightarrow{b} y \Downarrow_1 \]
Traces via determinisation

NFA

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

DFA

\[ \mathcal{P}X \rightarrow 2 \times (\mathcal{P}X)^A \]
Traces via determinisation

NFA

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

\[ \begin{align*}
  a, b \\
  x \downarrow_0 \quad b \quad y \downarrow_1
\end{align*} \]

\[ \begin{align*}
  x \xrightarrow{a} t_x, y \xrightarrow{a} t_y \\
  x \oplus y \xrightarrow{a} t_x \oplus t_y
\end{align*} \]

DFA

\[ \mathcal{P}X \rightarrow 2 \times (\mathcal{P}X)^A \]

\[ \begin{align*}
  a, a \\
  x \downarrow_0 \quad b \quad (x \oplus y) \downarrow_1
\end{align*} \]

\[ \begin{align*}
  x \downarrow o_x, y \downarrow o_y \\
  x \oplus y \downarrow o_x \oplus o_y
\end{align*} \]
Traces via determinisation

NFA

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

DFA

\[ \mathcal{P}X \rightarrow 2 \times (\mathcal{P}X)^A \]

\[ x \xrightarrow{a} t_x, y \xrightarrow{a} t_y \]
\[ x \oplus y \xrightarrow{a} t_x \oplus t_y \]

\[ x \downarrow o_x, y \downarrow o_y \]
\[ x \oplus y \downarrow o_x \oplus o_y \]
Traces via determinisation

NFA
\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

DFA
\[ \mathcal{P}X \rightarrow 2 \times (\mathcal{P}X)^A \]

\begin{align*}
    \begin{array}{c}
        a, b \\
        x \downarrow_0 \rightarrow b \rightarrow y \downarrow_1
    \end{array}
\end{align*}

\begin{align*}
    \begin{array}{c}
        a, a \\
        x \downarrow_0 \rightarrow b \rightarrow (x \oplus y) \downarrow_1
    \end{array}
\end{align*}

\[ x \xrightarrow{a} t_x, \quad y \xrightarrow{a} t_y \]
\[ x \oplus y \xrightarrow{a} t_x \oplus t_y \]

\[ x \downarrow o_x, \quad y \downarrow o_y \]
\[ x \oplus y \downarrow o_x \oplus o_y \]

Algebras for \( P \)

join semilattices with bottom
Traces via determinisation

In this paper, we propose a theory of trace semantics for systems with nondeterminism and probability. We introduce now the main abstract approach founded on algebras and coalgebras and inspired by automata theory, and study determinisation of NPLTS in a way that unifies the generalised determinisation [30], providing e.g. an abstract treatment of language equivalence for automata. We use the proposals of trace semantics in the literature [23], [24], [25], [26], and study from challenges [14], [15], [16], [17], [18], [19], [20], [21].

We call such approaches (NPLTS) that we use as a running example. The type of choice, modelled abstractly by a monad $\mathcal{M}$, is often linked to a concrete algebraic theory, the presentation of $\mathcal{M}$ as maps $X \rightarrow 2^X (\mathcal{P}X)^A$.

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 local-view semantics from the literature. We introduce now the main approach based on algebras and coalgebras.

In order to illustrate our approach, it is convenient to compare the resulting semantics to known definitions of trace semantics in the literature [23], [24], [25], [26].

We show how trace semantics for such processes can be recovered by instantiating a coalgebraic construction known together in the theory of traces for NPLTS.

The Theory of Traces for Systems with Nondeterminism and Probability

1. Introduction

In this paper, we propose a theory of trace semantics for systems with nondeterminism and probability.

To illustrate our approach, we use the nondeterministic probabilistic automaton (as depicted below) and we inductively compute $x \rightarrow a \rightarrow t_x$, $y \rightarrow a \rightarrow t_y$.

Having such a presentation is a valuable piece of our puzzle, and show how everything combines together in the theory of traces for NPLTS.

We define the probability of a trace as the sum of the probabilities of the individual actions making up the trace.

The interplay of nondeterminism and probability has been posing some remarkable challenges [14], [15], [16], [17], [18], [19], [20], [21].

At the same time, the interplay of nondeterminism and probability is abundantly used in verification [1].

AI [8], [9], [10], and studied from several decades.
Traces via determinisation

NFA

\[ X \rightarrow 2 \times (\mathcal{P}X)^A \]

DFA

\[ \mathcal{P}X \rightarrow 2 \times (\mathcal{P}X)^A \]

\[ a, b \]

\[ x \downarrow_0 \quad b \rightarrow \quad y \downarrow_1 \]

\[ x \rightarrow a \quad t_x, y \rightarrow a \quad t_y \]

\[ x \oplus y \rightarrow a \quad t_x \oplus t_y \]

Algebras for \( \mathcal{P} \)

finite powerset!

join semilattices with bottom

2 = \( \mathcal{P}1 \)
Traces via determinisation

Rabin PA

\[ X \rightarrow [0,1] \times (\mathcal{D}_{\leq 1} X)^A \]
Traces via determinisation

Rabin PA

\[ X \rightarrow [0,1] \times (\mathcal{P}_{\leq 1} X)^A \]
Traces via determinisation

Rabin PA
\[ X \rightarrow [0,1] \times (\mathcal{D}_{\leq 1}X)^A \]

DPA
\[ \mathcal{D}_{\leq 1}X \rightarrow [0,1] \times (\mathcal{D}_{\leq 1}X)^A \]
Traces via determinisation

Rabin PA

\[ X \rightarrow [0, 1] \times (\mathcal{D}_{\leq 1} X)^A \]

\[ x \downarrow_0 a, b \xrightarrow{\frac{1}{2}} y \downarrow_1 a, b \]

DPA

\[ \mathcal{D}_{\leq 1} X \rightarrow [0, 1] \times (\mathcal{D}_{\leq 1} X)^A \]

\[ x \downarrow_0 a, b \]
\[ (x + \frac{1}{2} y) \downarrow_\frac{1}{2} a, b \]
\[ (x + \frac{1}{4} y) \downarrow_\frac{3}{4} a, b \]
\[ \ldots \]

Algebras for \( \mathcal{D}_{\leq 1} \)
Traces via determinisation

Rabin PA
\[ X \rightarrow [0,1] \times (\mathcal{D}_{\leq 1}X)^A \]

DPA
\[ \mathcal{D}_{\leq 1}X \rightarrow [0,1] \times (\mathcal{D}_{\leq 1}X)^A \]

Algebras for \( \mathcal{D}_{\leq 1} \)

positive convex algebras

Having such a presentation is a valuable algebraic structure.
Traces via determinisation

Rabin PA

\[ X \rightarrow [0,1] \times (\mathcal{D}_{\leq 1}X)^A \]

- Algebras for \( \mathcal{D}_{\leq 1} \)
- Finitely supported subdistributions!

DPA

\[ \mathcal{D}_{\leq 1}X \rightarrow [0,1] \times (\mathcal{D}_{\leq 1}X)^A \]

- Positive convex algebras

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

Rabin PA

\[ X \rightarrow [0, 1] \times (\mathcal{D}_{\leq 1} X)^A \]

DPA

\[ \mathcal{D}_{\leq 1} X \rightarrow [0, 1] \times (\mathcal{D}_{\leq 1} X)^A \]

Algebras for \( \mathcal{D}_{\leq 1} \)

finitely supported subdistributions!

positive convex algebras

[0, 1] = \( \mathcal{D}_{\leq 1} \)
Traces via determinisation

Simple NPA

\[ X \to ? \times (\mathcal{E}X)^A \]
Traces via determinisation

Simple NPA

\[ X \to ? x (eX)^A \]
Traces via determinisation

Simple NPA

\[ X \rightarrow ?x (\mathcal{E}X)^A \]

DNPA

\[ \mathcal{E}X \rightarrow ?x (\mathcal{E}X)^A \]

From the abstract theory, we additionally get that:

\[ ?x (\mathcal{E}X)^A \]

\[ x_1 \]

\[ a \]

\[ x_1 \oplus (x_3 + \frac{1}{2}x_2) \]
Traces via determinisation

Simple NPA

\[ X \rightarrow ? \times (\varepsilon X)^A \]

\[ \frac{1}{2} \]
\[ \Delta_1 \]
\[ x \]
\[ x_1 \]
\[ b \]
\[ \frac{1}{2} \]
\[ \Delta_2 \]
\[ a \]
\[ \frac{1}{2} \]
\[ x_2 \]
\[ \frac{1}{2} \]
\[ x_3 \]

DNPA

\[ \varepsilon X \rightarrow ? \times (\varepsilon X)^A \]

\[ x_1 \]
\[ a \]
\[ x_1 \oplus (x_3 + \frac{1}{2} x_2) \]

Algebras for C
Traces via determinisation

Simple NPA

$X \rightarrow ? x (\varepsilon X)^A$

DNPA

$\varepsilon X \rightarrow ? x (\varepsilon X)^A$

Algebras for $C$

convex semilattices

$\frac{x_1}{2} a \Delta_1 \frac{1}{2} b \Delta_2 \frac{1}{2} \frac{x_2}{2} x_1 \varepsilon x_3 + \frac{1}{2} x_2$

Remarkably, necessity and convenience go hand in hand. The presentation for $C$ is somewhat known, although the monad $\{c\} = C$ as semilattice operation and standard techniques for nonempty convex bisimilarity $[7], [39]$. The equivalence is implied by the standard branching-bisimulations.

These theories give rise to three interesting algebras that are at the same time algebras that are at the same time algebras that are at the same time algebras that are at the same.

We develop a theory of traces for NPLTS using such algebras that are at the same time algebras that are at the same time algebras that are at the same time algebras that are at the same.

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 testing semantics.

Having put it to good use is part of our contribution which, for the NPLTS from Figure 1, is somewhat known, although the monad $\{c\} = C$ as semilattice operation; 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). The effectiveness of the bisimulation up-to techniques is shown 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 combinational properties. The equivalence is implied by the standard branching-bisimulations.
Traces via determinisation

Simple NPA

\[ X \rightarrow ? x (eX)^A \]

DNPA

\[ eX \rightarrow ? x (eX)^A \]

Algebras for C

convex semilattices

finitely generated convex sets of distr…
Traces via determinisation

Simple NPA

\[ X \rightarrow ? \times (\mathcal{E}X)^A \]

DNPA

\[ \mathcal{E}X \rightarrow ? \times (\mathcal{E}X)^A \]

\[ x_1 \]
\[ a \]
\[ x_1 + (x_3 + \frac{1}{2} x_2) \]

Algebras for \( C \)

\[ ? = C1 \]

convex semilattices

finitely generated convex sets of distributions
Presentation for $c$

Algebras for $c$

convex semilattices

finitely generated convex sets of distr…
Presentation for $c$

Algebras for $c$

convex semilattices

finitely generated convex sets of distr…

Bonchi, S., Vignudelli ‘19
Presentation for $c$

Algebras for $c$

convex semilattices

finitely generated convex sets of distr…

$\Delta = (A, \oplus, +_p)$

Bonchi, S., Vignudelli ‘19
Presentation for \( c \)

Algebras for \( c \)

convex semilattices

finitely generated convex sets of distr...

\[ \Delta = (A, \oplus, +p) \]

\( p \in (0, 1) \)

Bonchi, S., Vignudelli '19
Presentation for $\mathcal{C}$

Algebras for $\mathcal{C}$

convex semilattices

finitely generated convex sets of distributions

$p \in (0, 1)$

$\Delta = (A, \oplus, +p)$

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

$$\begin{align*}
(x +q y) +p z & \overset{(A_p)}{=} x +p q (y +p(1-q) z) \\
x +p y & \overset{(C_p)}{=} y +1-p x \\
x +p x & \overset{(I_p)}{=} x \\
\end{align*}$$

$$\begin{align*}
(x \oplus y) +p z & \overset{(D)}{=} (x +p z) \oplus (y +p z)
\end{align*}$$

The unit of $\mathcal{C}$ is a monad. Moreover, we would like to emphasise that having the presentation $\Delta$ on the convex closure of $(\mathcal{C}, M)$ is a consequence, as a consequence, of the following axiom. Consider the following algebraic theory (with somewhat different basic algebraic structure and nondeterminism starting from a similar algebraic structure). Then $\mathcal{C}$ is finitely generated convex algebras. Another presentation of convex algebras is given by the algebras for $\mathcal{C}$. Let $\mathcal{A}$ be a monad for the monad $\mathcal{C}$.

Bonchi, S., Vignudelli ‘19

Personal communication with Gordon Plotkin.
Presentation for $C$

Algebras for $C$

convex semilattices

finitely generated convex sets of distributions

$\mathbb{A} = (A, \oplus, +_p)$

$p \in (0, 1)$

Bonchi, S., Vignudelli ‘19

semilattice

\begin{align*}
(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{align*}

\begin{align*}
(x +_q y) +_p z & \overset{(A_p)}{=} x +_p (y +_p (1-q) z) \\
x +_p y & \overset{(C_p)}{=} y +_1 x \\
x +_p x & \overset{(I_p)}{=} x
\end{align*}

\begin{align*}
(x \oplus y) +_p z & \overset{(D)}{=} (x +_p z) \oplus (y +_p z)
\end{align*}
Presentation for \( \mathcal{C} \)

**Algebras for \( \mathcal{C} \)**

**convex semilattices**

**Ax** = \((A, \oplus, +_p)\)

\( p \in (0, 1) \)

**Bonchi, S., Vignudelli ‘19**

**semilattice**

**convex algebra**

\[
\begin{align*}
(x \oplus y) \oplus z \overset{(A)}{=} x \oplus (y \oplus z) \\
x \oplus y \overset{(C)}{=} y \oplus x \\
x \oplus x \overset{(I)}{=} x
\end{align*}
\]

\[
\begin{align*}
(x +_q y) +_p z \overset{(A_p)}{=} x +_p (y + p(1-q) z) \\
x +_p y \overset{(C_p)}{=} y +_{1-p} x \\
x +_p x \overset{(I_p)}{=} x
\end{align*}
\]

\[
(x \oplus y) +_p z \overset{(D)}{=} (x +_p z) \oplus (y +_p z)
\]
Algebras for $\mathcal{C}$

\[ \mathbb{A} = (A, \oplus, +_p) \]

finitely generated convex sets of distributions

$p \in (0, 1)$

convex semilattices

Bonchi, S., Vignudelli '19

Bonchi, S., Woracek '15, '17, '18

Semilattice

Convex algebra

\[
\begin{align*}
(x \oplus y) \oplus z & \overset{(A)}{=} x \oplus (y \oplus z) \\
x \oplus y & \overset{(C)}{=} y \oplus x \\
x \oplus x & \overset{(I)}{=} x
\end{align*}
\]

\[
\begin{align*}
(x +_q y) +_p z & \overset{(A_p)}{=} x +_p (y +_p (1 - p) z) \\
x +_p y & \overset{(C_p)}{=} y +_p x \\
x +_p x & \overset{(I_p)}{=} x
\end{align*}
\]

\[
(x \oplus y) +_p z \overset{(D)}{=} (x +_p z) \oplus (y +_p z)
\]
Presentation for \( \mathcal{C} \)

- **Algebras for \( \mathcal{C} \)**
- **Convex semilattices**
- **Bonacci, S., Vignudelli '19**
- **S., Woracek '15, '17, '18**
- **Semilattice**
- **Convex algebra**
- **Distributivity**

\[ \Delta = (A, \ominus, +p) \]

- \( p \in (0, 1) \)

\[
\begin{align*}
(x \oplus y) \oplus z & \overset{(A)}{=} x \oplus (y \oplus z) \\
x \oplus y & \overset{(C)}{=} y \oplus x \\
x \oplus x & \overset{(I)}{=} x
\end{align*}
\]

\[
\begin{align*}
(x +_q y) +_p z & \overset{(A_p)}{=} x +_p (y +_p (1-pq) z) \\
x +_p y & \overset{(C_p)}{=} y +_1 - p x \\
x +_p x & \overset{(I_p)}{=} x
\end{align*}
\]

\[
\begin{align*}
(x \oplus y) +_p z & \overset{(D)}{=} (x +_p z) \oplus (y +_p z)
\end{align*}
\]
Three variants for “C”
Three variants for “C”
Three variants for “$c$”

Bonchi, S., Vignudelli ‘19

We explore the whole space and prove coincidence with “local” trace semantics
Three variants for “$C$”

- Algebras for “$C$”
  - nonempty f.g. convex subsets of $\text{subdistr…}$

- Bonchi, S., Vignudelli ‘19
  - We explore the whole space and prove coincidence with “local” trace semantics
Three variants for “$c$”

Algebras for “$c$”

nonempty f.g. convex subsets of subdistr…

Bonchi, S., Vignudelli ‘19
We explore the whole space and prove coincidence with “local” trace semantics

I.pointed convex semilattices

Intervals in $[0,1]$ with min-max Minkowski $[0,0] = “c”1$
Three variants for “$C$”

I. Pointed convex semilattices

- Intervals in $[0,1]$ with min-max Minkowski $[0,0] = "C"1$

II. With bottom

- $[0,1]$ with max, $+_p$
- $0 = "C"1$

Bonchi, S., Vignudelli ‘19

We explore the whole space and prove coincidence with “local” trace semantics

Algebras for “$C$”

- Nonempty f.g. convex subsets of subdistr…
Three variants for “C”

I. pointed convex semilattices
- Intervals in [0,1] with min-max
- Minkowski [0,0] = “C”1

II. with bottom
- [0,1] with max, +p
- 0 = “C”1

III. with top
- [0,1] with min, +p
- 0 = “C”1

Bonchi, S., Vignudelli ‘19

We explore the whole space and prove coincidence with “local” trace semantics
Three things to take home:

1. Semantics via determinisation is easy for automata with M-effects

2. Having a presentation for M gives us syntax

3. Having the syntax makes determinisation natural!
Three things to take home:

1. Semantics via determinisation is easy for automata with M-effects

2. Having a presentation for M gives us syntax

3. Having the syntax makes determinisation natural!

Many general properties follow also a sound up-to context proof technique.
Three things to take home:

1. Semantics via determinisation is easy for automata with M-effects

2. Having a presentation for M gives us syntax

3. Having the syntax makes determinisation natural!

Many general properties follow also a sound up-to context proof technique.

Combining nondeterminism and probability becomes easy.
Three things to take home:

1. Semantics via determinisation is easy for automata with M-effects

2. Having a presentation for M gives us syntax

3. Having the syntax makes determinisation natural!

Many general properties follow also a sound up-to context proof technique.