Trace semantics of well-founded processes via commutativity

For Gordon Plotkin on his 70th birthday

Nathan Bowler (Hamburg) and Paul Blain Levy (Birmingham)

November 23, 2016

Outline

An imperative language

$$
\begin{array}{rcl}
M & ::= & \mathtt{input}_{\mathsf{age}}(M_n)_{n \in \mathbb{N}} \\
& & \mid \mathtt{input}_{\mathsf{happy}}(M_{\mathsf{yes}}, M_{\mathsf{no}}) \\
& & \mid \mathtt{input}_{\mathsf{bye}}() \mid M \ \mathtt{or} \ M
\end{array}
$$

An imperative language

$$
\begin{aligned}
M \quad ::= \quad & \mathtt{input}_{\mathsf{age}}(M_n)_{n\in\mathbb{N}} \\
& \mid \mathtt{input}_{\mathsf{happy}}(M_{\mathsf{yes}}, M_{\mathsf{no}}) \\
& \mid \mathtt{input}_{\mathsf{bye}}() \mid M \ \mathtt{or}\ M
\end{aligned}
$$

- $\mathtt{input}_{\mathsf{age}}(M_n)_{n\in\mathbb{N}}$ prints `What is your age?` and pauses.
  - If the user enters n, it proceeds to execute M_n.

An imperative language

$$
\begin{aligned}
M \quad ::= \quad & \mathtt{input}_{\text{age}}(M_n)_{n\in\mathbb{N}} \\
& \mid \mathtt{input}_{\text{happy}}(M_{\text{yes}}, M_{\text{no}}) \\
& \mid \mathtt{input}_{\text{bye}}() \mid M \ \mathtt{or}\ M
\end{aligned}
$$

- $\mathtt{input}_{\text{age}}(M_n)_{n\in\mathbb{N}}$ prints `What is your age?` and pauses.
  - If the user enters n, it proceeds to execute M_n.
- $\mathtt{input}_{\text{happy}}(M_{\text{yes}}, M_{\text{no}})$ prints `Are you happy?` and pauses.
  - If the user enters `Yes`, it executes M_{yes}.
  - If the user enters `No`, it executes M_{no}.

An imperative language

$$M \quad ::= \quad \mathtt{input}_{\mathsf{age}}(M_n)_{n\in\mathbb{N}} \mid \mathtt{input}_{\mathsf{happy}}(M_{\mathsf{yes}}, M_{\mathsf{no}}) \mid \mathtt{input}_{\mathsf{bye}}() \mid M \ \mathtt{or}\ M$$

- $\mathtt{input}_{\mathsf{age}}(M_n)_{n\in\mathbb{N}}$ prints `What is your age?` and pauses.
  - If the user enters n, it proceeds to execute M_n.
- $\mathtt{input}_{\mathsf{happy}}(M_{\mathsf{yes}}, M_{\mathsf{no}})$ prints `Are you happy?` and pauses.
  - If the user enters `Yes`, it executes M_{yes}.
  - If the user enters `No`, it executes M_{no}.
- $\mathtt{input}_{\mathsf{bye}}()$ prints `Goodbye` and pauses.

An imperative language

$$M \quad ::= \quad \mathtt{input}_{\mathsf{age}}(M_n)_{n\in\mathbb{N}} \mid \mathtt{input}_{\mathsf{happy}}(M_{\mathsf{yes}}, M_{\mathsf{no}}) \mid \mathtt{input}_{\mathsf{bye}}() \mid M \ \mathtt{or}\ M$$

- $\mathtt{input}_{\mathsf{age}}(M_n)_{n\in\mathbb{N}}$ prints `What is your age?` and pauses.
  - If the user enters n, it proceeds to execute M_n.
- $\mathtt{input}_{\mathsf{happy}}(M_{\mathsf{yes}}, M_{\mathsf{no}})$ prints `Are you happy?` and pauses.
  - If the user enters `Yes`, it executes M_{yes}.
  - If the user enters `No`, it executes M_{no}.
- $\mathtt{input}_{\mathsf{bye}}()$ prints `Goodbye` and pauses.
- $M\ \mathtt{or}\ N$ nondeterministically chooses to execute M or N.

Signature

An I/O signature consists of

- a set K of operations

$$\textit{Example} \quad K \quad = \quad \{\mathrm{age}, \mathrm{happy}, \mathrm{bye}\}$$

Signature

An I/O signature consists of

- a set K of operations

$$\textit{Example} \quad K = \{\mathrm{age}, \mathrm{happy}, \mathrm{bye}\}$$

- for each operation k, a set $\mathrm{Ar}(k)$ of argument indices, called the arity of k.

$$\textit{Example} \quad \begin{aligned} \mathrm{Ar}(\mathrm{age}) &= \mathbb{N} \\ \mathrm{Ar}(\mathrm{happy}) &= \{\mathrm{yes}, \mathrm{no}\} \\ \mathrm{Ar}(\mathrm{bye}) &= \emptyset \end{aligned}$$

Signature

An I/O signature consists of

- a set K of operations

$$\textit{Example} \quad K = \{\text{age}, \text{happy}, \text{bye}\}$$

- for each operation k, a set $\text{Ar}(k)$ of argument indices, called the arity of k.

$$\begin{array}{lrcl} \textit{Example} & \text{Ar}(\text{age}) & = & \mathbb{N} \\ & \text{Ar}(\text{happy}) & = & \{\text{yes}, \text{no}\} \\ & \text{Ar}(\text{bye}) & = & \emptyset \end{array}$$

Each signature $S = (\text{Ar}(k))_{k \in K}$ gives rise to a language:

$$M ::= \ \mathtt{input}_k(M_i)_{i \in \text{Ar}(k)} \mid M \ \mathtt{or} \ M$$

Operational semantics

Write Comm for the set of commands.

Operational semantics is a function $\zeta \ : \ \mathrm{Comm} \to \mathcal{P}_{\mathrm{f}}^{+} \sum_{k \in K} \mathrm{Comm}^{\mathrm{Ar}(k)}$.

where $\mathcal{P}_{\mathrm{f}}^{+} X$ is the set of nonempty finite subsets of X.

Operational semantics

Write Comm for the set of commands.

Operational semantics is a function $\zeta : \text{Comm} \to \mathcal{P}_\mathsf{f}^+ \sum_{k \in K} \text{Comm}^{\text{Ar}(k)}$.

where $\mathcal{P}_\mathsf{f}^+ X$ is the set of nonempty finite subsets of X.

More succinctly, (Comm, ζ) is a $\mathcal{P}_\mathsf{f}^+ H^S$-coalgebra

where $H^S : Y \mapsto \sum_{k \in K} Y^{\text{Ar}(k)}$.

Traces of a command?

A play is a sequence $k_0, i_0, k_1, i_1, \ldots$, where for all n

- $k_n \in K$
- $i_n \in \mathrm{Ar}(k_n)$.

It can be active-ending (even length), passive-ending (odd length) or infinite.

Traces of a command?

A play is a sequence $k_0, i_0, k_1, i_1, \ldots$, where for all n

- $k_n \in K$
- $i_n \in \mathsf{Ar}(k_n)$.

It can be active-ending (even length), passive-ending (odd length) or infinite.

A play is a trace of a command M when

there exists a sequence $u_0, \ldots, u_n$, with $u_r \in \mathsf{Comm}^{\mathsf{Ar}(k_r)}$,

such that $(k_0, u_0) \in \zeta M_0 \wedge \cdots \wedge (k_n, u_n) \in \zeta M_n$

writing $M_0 \stackrel{\text{def}}{=} M$ and $M_{r+1} \stackrel{\text{def}}{=} u_r i_r$.

No infinite traces

No command has an infinite trace $k_0, i_0, k_1, i_1, \ldots$

No infinite traces

No command has an infinite trace $k_0, i_0, k_1, i_1, \ldots$

(Comm, ζ) is a well-founded $\mathcal{P}_{\mathrm{f}}^{+} H^{S}$-coalgebra (Taylor).

Bisimilarity and traces

Two important equivalence relations on commands

More generally: on states of $\mathcal{P}_{\mathrm{f}}^{+} H^{S}$-coalgebras.

Commands M and N are

- bisimilar when there is a bisimulation that relates them.
- trace equivalent when they have the same traces.

Bisimilarity implies trace equivalence.

Play processes

A play process is a set D of passive-ending plays that is prefix-closed:

- $tki \in D$ implies $t \in D$

Let D^+ be the corresponding set of enabled active-ending plays:

$$D^+ = \{\varepsilon\} \cup \{ski \mid sk \in D,\, i \in \mathrm{Ar}(k)\}$$

Play processes

A play process is a set D of passive-ending plays that is prefix-closed:

- $tki \in D$ implies $t \in D$

Let D^+ be the corresponding set of enabled active-ending plays:

$$D^+ = \{\varepsilon\} \cup \{ski \mid sk \in D,\ i \in \mathrm{Ar}(k)\}$$

A response to $t \in D^+$ is $k \in K$ such that $tk \in D$.

Play processes

A play process is a set D of passive-ending plays that is prefix-closed:

- $tki \in D$ implies $t \in D$

Let D^+ be the corresponding set of enabled active-ending plays:

$$D^+ = \{\varepsilon\} \cup \{ski \mid sk \in D,\ i \in \mathrm{Ar}(k)\}$$

A response to $t \in D^+$ is $k \in K$ such that $tk \in D$.

The trace set of any command is a play process.

Play processes

A play process is a set D of passive-ending plays that is prefix-closed:

- $tki \in D$ implies $t \in D$

Let D^+ be the corresponding set of enabled active-ending plays:

$$D^+ = \{\varepsilon\} \cup \{ski \mid sk \in D,\ i \in \mathrm{Ar}(k)\}$$

A response to $t \in D^+$ is $k \in K$ such that $tk \in D$.

The trace set of any command is a play process.

More generally, the trace set of any state of a $\mathcal{P}H^S$-coalgebra.

Two questions

1. Under what conditions is a play process D the trace set of a command?

Two questions

1. Under what conditions is a play process D the trace set of a command?
2. Can we give an axiomatic theory of trace equivalent commands?

Tree = deterministic process

A tree is a play process D such that every $t \in D^+$ has a unique response.

Tree = deterministic process

A tree is a play process D such that every $t \in D^+$ has a unique response.

D^∞ is the set of infinite branches, i.e. infinite plays whose passive-ending prefixes are all in D.

If this set is empty, D is well-founded.

Tree = deterministic process

A tree is a play process D such that every $t \in D^+$ has a unique response.

D^∞ is the set of infinite branches, i.e. infinite plays whose passive-ending prefixes are all in D.

If this set is empty, D is well-founded.

A play process is the trace set of a deterministic command iff it is a well-founded tree.
Deterministic commands M and M' have the same trace set iff $M = M'$.

S-algebras

An S-algebra consists of

- a set X
- for each operation $k \in K$, a function

$$\mathrm{op}_k \ : \ X^{\mathrm{Ar}(k)} \to X$$

S-algebras

An S-algebra consists of

- a set X
- for each operation $k \in K$, a function

$$\mathrm{op}_k \ : \ X^{\mathrm{Ar}(k)} \to X$$

Famous theorem

The set of well-founded trees is an initial object in the category of S-algebras.

Definable play processes (Plotkin)

A play process D is

- *total* when every $t \in D^+$ has at least one response
- *finitely nondeterministic* when every $t \in D^+$ has only finitely many responses; we then write D^∞ for the set of infinite plays whose prefixes are all in D
- *König* when it is finitely nondeterministic and D^∞ is empty.

Definable play processes (Plotkin)

A play process D is

- *total* when every $t \in D^+$ has at least one response
- *finitely nondeterministic* when every $t \in D^+$ has only finitely many responses; we then write D^∞ for the set of infinite plays whose prefixes are all in D
- *König* when it is finitely nondeterministic and D^∞ is empty.

The trace set of any command is total and König.

Definable play processes (Plotkin)

A play process D is

- *total* when every $t \in D^+$ has at least one response
- *finitely nondeterministic* when every $t \in D^+$ has only finitely many responses; we then write D^∞ for the set of infinite plays whose prefixes are all in D
- *König* when it is finitely nondeterministic and D^∞ is empty.

The trace set of any command is total and König.

More generally, the trace set of any well-founded $\mathcal{P}_\mathrm{f}^+ H^S$-coalgebra.

Definable play processes (Plotkin)

A play process D is

- *total* when every $t \in D^+$ has at least one response
- *finitely nondeterministic* when every $t \in D^+$ has only finitely many responses; we then write D^∞ for the set of infinite plays whose prefixes are all in D
- *König* when it is finitely nondeterministic and D^∞ is empty.

The trace set of any command is total and König.

More generally, the trace set of any well-founded $\mathcal{P}_{\mathrm{f}}^{+} H^{S}$-coalgebra.

Conversely, any total and König play process is the trace set of a command.

Basic axioms

The `or` operation is idempotent, commutative and associative.

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

Basic axioms

The `or` operation is idempotent, commutative and associative.

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

This characterizes bisimilarity.

What axioms should we add for trace equivalence?

Commuting operations

Let X be a set equpped with operations $f \; : \; X^I \to X$ and $g \; : \; X^J \to X$. These operations commute when

$$f(g(x_{i,j})_{j \in J})_{i \in I} \quad = \quad g(f(x_{i,j})_{i \in I})_{j \in J}$$

for any $I \times J$-matrix $(x_{i,j})_{i \in I, j \in J}$ of elements of X.

I/O commutes with nondeterminism

For trace equivalence, the or operation commutes with $\texttt{input}_k$ for each $k \in K$.

$$\texttt{input}_k(M_i \ \texttt{or}\ N_i)_{i \in \mathsf{Ar}(k)} \quad = \quad \texttt{input}_k(M_i)_{i \in \mathsf{Ar}(k)} \ \texttt{or}\ \texttt{input}_k(N_i)_{i \in \mathsf{Ar}(k)}$$

I/O commutes with nondeterminism

For trace equivalence, the `or` operation commutes with $\mathtt{input}_k$ for each $k \in K$.

$$\mathtt{input}_k(M_i \ \mathtt{or}\ N_i)_{i \in \mathsf{Ar}(k)} \quad = \quad \mathtt{input}_k(M_i)_{i \in \mathsf{Ar}(k)} \ \mathtt{or}\ \mathtt{input}_k(N_i)_{i \in \mathsf{Ar}(k)}$$

Note: this follows from idempotency if k is a constant.

I/O commutes with nondeterminism

For trace equivalence, the $\mathtt{or}$ operation commutes with $\mathtt{input}_k$ for each $k \in K$.

$$\mathtt{input}_k(M_i \ \mathtt{or}\ N_i)_{i \in \mathrm{Ar}(k)} \quad = \quad \mathtt{input}_k(M_i)_{i \in \mathrm{Ar}(k)} \ \mathtt{or}\ \mathtt{input}_k(N_i)_{i \in \mathrm{Ar}(k)}$$

Note: this follows from idempotency if k is a constant.

We now have a sound and complete axiomatic theory. (Plotkin)

Commuting semilattice algebras

A semilattice is

- a set X equipped with an idempotent, commutative and associative operation $\vee$
- or equivalently: a poset $(X, \leqslant)$ with binary joins.

Commuting semilattice algebras

A semilattice is

- a set X equipped with an idempotent, commutative and associative operation $\vee$
- or equivalently: a poset $(X, \leqslant)$ with binary joins.

An semilattice S-algebra is a semilattice $(X, \vee)$ equipped with operations $\mathrm{op}_k \ : \ X^{\mathrm{Ar}(k)} \to X$ for all $k \in K$.
Not necessarily monotone.

Commuting semilattice algebras

A semilattice is

- a set X equipped with an idempotent, commutative and associative operation $\vee$
- or equivalently: a poset $(X, \leqslant)$ with binary joins.

An semilattice S-algebra is a semilattice $(X, \vee)$ equipped with operations $\mathsf{op}_k \ : \ X^{\mathsf{Ar}(k)} \to X$ for all $k \in K$.
Not necessarily monotone.

An semilattice S-algebra is commuting when for all $k \in K$ it satisfies

$$\mathsf{op}_k(x_i \vee y_i)_{i \in \mathsf{Ar}(k)} \quad = \quad \mathsf{op}_k(x_i)_{i \in \mathsf{Ar}(k)} \vee \mathsf{op}_k(y_i)_{i \in \mathsf{Ar}(k)}$$

This implies that op_k is monotone.

Summary: universal property

The set of total and König play processes

Summary: universal property

The set of total and König play processes

is an initial object in the category of commuting semilattice S-algebras.

Countable nondeterminism

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

Operational semantics is a function $\zeta \ : \ \mathsf{Comm} \to \mathcal{P}_{\mathsf{c}}^{+} \sum_{k \in K} \mathsf{Comm}^{\mathsf{Ar}(k)}$.

Countable nondeterminism

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

Operational semantics is a function $\zeta : \mathsf{Comm} \to \mathcal{P}_{\mathsf{c}}^{+} \sum_{k \in K} \mathsf{Comm}^{\mathsf{Ar}(k)}$.

1. Under what conditions is a play process D the trace set of a command?
2. Can we give an axiomatic theory of trace equivalent commands?

Definable play processes

A play process D is

- *countably nondeterministic* when every $t \in D^+$ has only countably many responses
- *well-foundedly total* when for all $t \in D^+$ there is a well-founded tree E such that $\{ts \mid s \in E\} \subseteq D$.

Definable play processes

A play process D is

- *countably nondeterministic* when every $t \in D^+$ has only countably many responses
- *well-foundedly total* when for all $t \in D^+$ there is a well-founded tree E such that $\{ts \mid s \in E\} \subseteq D$.

The trace set of any command has these properties.

More generally, the trace set of any well-founded $\mathcal{P}_c^+ H^S$-coalgebra.

Definable play processes

A play process D is

- *countably nondeterministic* when every $t \in D^+$ has only countably many responses
- *well-foundedly total* when for all $t \in D^+$ there is a well-founded tree E such that $\{ts \mid s \in E\} \subseteq D$.

The trace set of any command has these properties.

More generally, the trace set of any well-founded $\mathcal{P}_\mathrm{c}^+ H^S$-coalgebra.

Conversely, any play process with these properties is the trace set of a command.

Basic axioms

The `or` operation is idempotent, commutative and associative.

The `choose` operation is idempotent and extrusive, and `or` distributes over it.

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

Basic axioms

The or operation is idempotent, commutative and associative.

The choose operation is idempotent and extrusive, and or distributes over it.

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

This characterizes bisimilarity.

What axioms should we add for trace equivalence?

I/O commutes with nondeterminism

For trace equivalence, the $\texttt{choose}$ operation commutes with $\texttt{input}_k$ for each $k \in K$.

$$\texttt{input}_k((\texttt{choose}\ (M_{i,n})_{n\in\mathbb{N}})_{i\in\mathsf{Ar}(k)} \quad = \quad \texttt{choose}\ (\texttt{input}_k(M_{i,n})_{i\in\mathsf{Ar}(k)})_{n\in\mathbb{N}}$$

Note: this follows from idempotency if k is a constant.

We now have a sound and complete axiomatic theory.

Finite probabilistic choice

$$M ::= \quad \texttt{input}_k(M_i)_{i \in \mathsf{Ar}(k)} \mid M +_p M \quad (p \in (0,1))$$

Operational semantics is a function $\zeta : \mathsf{Comm} \to \mathcal{D}_\mathsf{f} \sum_{k \in K} \mathsf{Comm}^{\mathsf{Ar}(k)}$.

Probabilistic play processes

A probabilistic play process is a function μ from plays to $[0, 1]$ such that for every active-ending play t we have

$$\mu^+(t) = \sum_{k \in K} \mu(tk)$$

where we write

$$\begin{array}{rcl} \mu^+(\varepsilon) & = & 1 \\ \mu^+(ti) & = & \mu(t) \end{array}$$

Probabilistic play processes

A probabilistic play process is a function μ from plays to $[0, 1]$ such that for every active-ending play t we have

$$\mu^+(t) = \sum_{k \in K} \mu(tk)$$

where we write

$$\begin{aligned} \mu^+(\varepsilon) &= 1 \\ \mu^+(ti) &= \mu(t) \end{aligned}$$

Any command gives a probabilistic play process.

More generally, any state of a $\mathcal{D}H^S$-coalgebra.

Probabilistic play processes

A probabilistic play process is a function μ from plays to $[0,1]$ such that for every active-ending play t we have

$$\mu^+(t) = \sum_{k \in K} \mu(tk)$$

where we write

$$\begin{array}{rcl} \mu^+(\varepsilon) & = & 1 \\ \mu^+(ti) & = & \mu(t) \end{array}$$

Any command gives a probabilistic play process.

More generally, any state of a $\mathcal{D}H^S$-coalgebra.

The support of μ is the set of passive-ending plays t with $\mu(t) > 0$.
It is a total and countably nondeterministic play process.

Our questions again

When is a probabilistic play process μ definable by a command?

- When the support of μ is König.

What must we add to the basic axioms of probabilistic choice to obtain a theory of trace equivalence?

- Commutativity of $+_p$ with $\mathtt{input}_k$.

Countable probabilistic choice

$$
\begin{array}{rcl}
M & ::= & \mathtt{input}_k(M_i)_{i \in \mathsf{Ar}(k)} \mid M +_p M \quad (p \in (0,1)) \\
 & \mid & \sum_{n \in \mathbb{N}} p_n M_n \quad (p_n > 0, \sum_{n \in \mathbb{N}} p_n = 1)
\end{array}
$$

Operational semantics is a function $\zeta : \mathsf{Comm} \to \mathcal{D} \sum_{k \in K} \mathsf{Comm}^{\mathsf{Ar}(k)}$.

Counterstrategies

A set C of active-ending plays is a counterstrategy when:

- It is prefix-closed.
- It contains the empty play.
- Every $s \in C^+$ has at most one response.

Write C^{Fail} for the set of failures of C, i.e. $s \in C^+$ with no response.

Failing against a play process

Let C be a counterstrategy.

- For a tree D
  there's a unique play in $(D \cap C^{\mathsf{Fail}}) \cup (D^{\infty} \cap C^{\infty})$.
  If it's in the left part, C fails against D.
- C frequently fails against a play process D
  when every play in $D^{+} \cap C$ extends to a play in $D \cap C^{\mathsf{Fail}}$.
- C almost surely fails against a probabilistic play process μ when

$$\sum_{s \in D \cap C^{\mathsf{Fail}}} \mu(s) = 1.$$

All counterstrategies failing

- A tree is well-founded
  iff every counterstrategy fails against it.
- A play process is well-foundedly total
  iff every counterstrategy frequently fails against it.
- A probabilistic play process is victorious
  when every counterstrategy almost surely fails against it.

All counterstrategies failing

- A tree is well-founded
  iff every counterstrategy fails against it.
- A play process is well-foundedly total
  iff every counterstrategy frequently fails against it.
- A probabilistic play process is victorious
  when every counterstrategy almost surely fails against it.

Definable = victorious.

Commutativity?

What must we add to the basic axioms of probabilistic choice to obtain a theory of trace equivalence?

Commutativity of $+_p$ and $\sum_{n\in\mathbb{N}} p_n$ with $\mathtt{input}_k$ gives a sound logic.

But is it complete?

Commutativity?

What must we add to the basic axioms of probabilistic choice to obtain a theory of trace equivalence?

Commutativity of $+_p$ and $\sum_{n \in \mathbb{N}} p_n$ with $\mathtt{input}_k$ gives a sound logic.

But is it complete?

This is an open problem,
even for the signature consisting of a binary operation and a constant.

Conclusions

- In all cases we have characterized the definable play processes.

Conclusions

- In all cases we have characterized the definable play processes.
- In all cases commutativity gives a sound logic of trace equivalence.
- It is complete for finite and countable nondeterministic choice, and finite probabilistic choice.

Conclusions

- In all cases we have characterized the definable play processes.
- In all cases commutativity gives a sound logic of trace equivalence.
- It is complete for finite and countable nondeterministic choice, and finite probabilistic choice.
- In these cases the set of play processes satisfying the appropriate condition is an initial object in the appropriate category of commuting algebras.