Probabilistic Petri Nets, Event Structures and Domains

Daniele Varacca
with Mogens Nielsen, and Glynn Winskel

ENS - Paris
BRICS - Aarhus

IRISA - Rennes, November 7th 2003
Thema

Probability and Concurrency
Road Map

- Interleaving
- Petri nets and Mazurkiewicz equivalence
- Event structures and tests
- Domains and valuations
Road Map

- Interleaving
  - Markov decision processes
  - Scheduler
- Petri nets and Mazurkiewicz equivalence
- Event structures and tests
- Domains and valuations
Probabilistic interleaving

Probability and concurrency

\[ s \rightarrow s' \]

\[ s \parallel t \rightarrow s' \parallel t \]
We want an associative parallel composition

Probability and concurrency

\[ s \xrightarrow{p} s' \]

\[ s \parallel t \xrightarrow{?} s' \parallel t \]
Probabilistic interleaving

Probability and concurrency

\[ s \xrightarrow{p} s' \]

\[ s \parallel t \xrightarrow{?} s' \parallel t \]

- We want an associative parallel composition
- We don’t want to commit to any particular scheduling policy

Usual solution:

combining nondeterminism and probabilities
Markov decision processes

$1\frac{1}{2}$ player game (Henzinger, Jurdzinski)

The moves of the full player are called actions
Scheduler

A scheduler is a strategy for the (full) player.
Scheduler
A scheduler is a strategy for the (full) player.
Scheduler

A scheduler is a strategy for the (full) player.

A scheduler leaves us with a (labelled) Markov process.
Scheduler

A scheduler is a strategy for the (full) player.
Scheduler

A scheduler is a strategy for the (full) player.

A scheduler leaves us with a (labelled) Markov process.
A probabilistic scheduler can use a mixed strategy.
A probabilistic scheduler can use a mixed strategy.
Probabilistic scheduler

A probabilistic scheduler can use a mixed strategy.
Probabilistic scheduler

A probabilistic scheduler can use a mixed strategy.
Runs

- A finite run of a Markov decision process is a probability distribution over strings of the same length (DeAlfaro, Henzinger, Jhala)
- The set of maximal runs is equipped with a measure (Segala)
- Probabilistic verification uses interleaving
- Temporal logics talk about schedulers
Parallel composition

- Parallel composition is modelled by interleaving
- The scheduling policy is left to the ..er.. scheduler!

Issues:

- We don’t distinguish nondeterminism due to scheduling from “genuine” nondeterminism
- The state space is big!
Road Map

- Interleaving
  - Markov decision processes
  - Scheduler
- Petri nets and Mazurkiewicz equivalence
- Event structures and tests
- Domains and valuations
Road Map

- Interleaving
- Petri nets and Mazurkiewicz equivalence
- Event structures and tests
- Domains and valuations
Road Map

- Interleaving
- Petri nets and Mazurkiewicz equivalence
  - Petri nets
  - Confusion
  - Probabilistic Petri nets
  - Mazurkiewicz equivalence
- Event structures and tests
- Domains and valuations
True concurrency

Keep track of independence and causality

- Transition systems with independence
- Event structures
- Petri nets

Occurrence nets $\leftrightarrow$ Event structures
Petri Nets

- Definitions: places, transitions, arcs, marking, enabling, etc
- Direct conflict: two transitions are in direct conflict if their neighborhoods are not disjoint
- Dynamic conflict: two transitions are in dynamic conflict at a marking $M$ if they are enabled at $M$ and in direct conflict

The conflict set of an enabled transition $t$ at a marking $M$ is the set of transitions which are in dynamic conflict with $t$
Petri Nets: Examples

A Petri Net. $b$ and $c$ are in direct conflict
Petri Nets: Examples

A marking: $b$ and $c$ are in dynamic conflict
Petri Nets: Examples

The marking after firing $a$

$d$ is not enabled
The marking after firing $b$

$d$ is enabled, $c$ is disabled
$a$ and $b$ are concurrent and can fire in any order
Petri Nets: Examples

\[ a \text{ and } b \text{ are concurrent and can fire in any order} \]
Petri Nets: Examples

$a$ and $b$ are concurrent and can fire in any order
Petri Nets: Examples

$a$ and $b$ can also fire together
Petri Nets: Examples

$a$ and $b$ can also fire together
Petri Nets: Examples

$b$ and $c$ are in direct conflict
but they are not in dynamic conflict at this marking
Petri Nets: Examples

after $a$ is fired, $b$ and $c$ are in dynamic conflict
Confusion

Crucial notion
A marking $M$ is confused at a transition $t$ if the firing of a concurrent transition $s$ changes the conflict set of $t$.
Confusion

Crucial notion
A marking $M$ is confused at a transition $t$ if the firing of a concurrent transition $s$ changes the conflict set of $t$

- Symmetric confusion: dynamic conflict is not transitive
- Asymmetric confusion: “dynamic and static conflict don’t coincide”
Confusion: Examples

Symmetric confusion
Confusion: Examples

Symmetric confusion
The firing of $a$ changes the conflict set of $c$
Confusion: Examples

Symmetric confusion
And symmetrically
Confusion: Examples

Asymmetric confusion
Confusion: Examples

Asymmetric confusion
The firing of $a$ changes the conflict set of $c$
Confusion free nets

Confusion freeness is a dynamic notion
Some static conditions can characterize it
Example: (extended) free choice
Petri Nets as MDP

Adding probabilities
Idea: restrict to Nets where at every marking the
dynamic conflict is an equivalence (= no symmetric
confusion)
The equivalence classes are the actions of the MDP
The choice within one class can be resolved
probabilistically
Petri Nets as MDP

Adding probabilities

Idea: restrict to Nets where at every marking the dynamic conflict is an equivalence (= no symmetric confusion)

The equivalence classes are the actions of the MDP

The choice within one class can be resolved probabilistically

To this aim:

- Every transition has a weight
- Probability is assigned by normalizing the weights within an action
Petri Nets as MDP - Examples

Two actions: \{a\}, \{c\}.
If the scheduler chooses \( \{a\} \)
Petri Nets as MDP - Examples

$\alpha$ is fired with probability 1
Petri Nets as MDP - Examples

Only one action.
Petri Nets as MDP - Examples

\[ c \text{ is fired with probability } \frac{1}{2} \]
Petri Nets as MDP - Examples

If the scheduler chooses \{c\}
There is only one (maximal) run
Petri Nets as MDP - Examples

There is only one (maximal) run
Finite runs - Examples

The first scheduler produces the following (maximal) distribution

\begin{align*}
ab & \mapsto \frac{1}{2} \\
ac & \mapsto \frac{1}{2}
\end{align*}

The second scheduler

\begin{align*}
ca & \mapsto 1
\end{align*}
The confusion free case
The confusion free case

The scheduler chooses
The confusion free case

\[ \spadesuit = \frac{1}{3}, \heartsuit = \frac{2}{3} \]
The confusion free case

\[ \spadesuit = \frac{2}{3} \]
The confusion free case

\[ \spadesuit = \frac{2}{3} \]
The confusion free case

\[ \spadesuit = \frac{2}{3} \]
The confusion free case

\[ \clubsuit = \frac{1}{3} \]
The confusion free case

\[ \clubsuit = \frac{1}{3} \]
The confusion free case

\[ \clubsuit = \frac{1}{3} \]
The confusion free case

I could have chosen a different action to start with
The confusion free case
The confusion free case

Probabilistic Petri Nets, Event Structures and Domains – p.20
The confusion free case
Finite runs - Examples

The first scheduler produces the following (maximal) distribution

\[
\begin{align*}
bad & \mapsto \frac{1}{3} \\
cea & \mapsto \frac{2}{3}
\end{align*}
\]

The second scheduler

\[
\begin{align*}
abd & \mapsto \frac{1}{3} \\
ace & \mapsto \frac{2}{3}
\end{align*}
\]
Mazurkiewicz equivalence

Concurrent alphabet \((\Sigma, \bowtie)\), where \(\bowtie\) is irreflexive and symmetric

For two strings, define

\[ \sigma \equiv \tau \]

if they are equal up to swapping of concurrent symbols
Mazurkiewicz equivalence

Concurrent alphabet \((\Sigma, \bowtie)\), where \(\bowtie\) is irreflexive and symmetric

For two strings, define

\[
\sigma \equiv \tau
\]

if they are equal up to swapping of concurrent symbols

Two probability distributions on strings are Mazurkiewicz equivalent if they assign the same probability to every equivalence class.
Mazurkiewicz - Examples

\[ a \bowtie b, c, e; b \bowtie e; c \bowtie d; d \bowtie e \]
Mazurkiewicz - Examples

\[ a \bowtie b, c, e; \ b \bowtie e; \ c \bowtie d; \ d \bowtie e \]

\[
\begin{align*}
bad & \mapsto \frac{1}{3} \\
cea & \mapsto \frac{2}{3}
\end{align*}
\]

The run obtained by the first scheduler
Mazurkiewicz - Examples

\[ a \bowtie b, c, e; \ b \bowtie e; \ c \bowtie d; \ d \bowtie e \]

\[
\begin{align*}
\hat{bad} & \mapsto \frac{1}{3} \\
\hat{cea} & \mapsto \frac{2}{3}
\end{align*}
\]
Mazurkiewicz - Examples

\[ a \bowtie b, c, e; b \bowtie e; c \bowtie d; d \bowtie e \]

\[ \begin{align*}
  \text{bad} & \mapsto \frac{1}{3} \\
  \text{cea} & \mapsto \frac{2}{3}
\end{align*} \]

\[ \begin{align*}
  \text{abd} & \mapsto \frac{1}{3} \\
  \text{cae} & \mapsto \frac{2}{3}
\end{align*} \]
Mazurkiewicz - Examples

\[ a \bowtie b, c, e; b \bowtie e; c \bowtie d; d \bowtie e \]

\[
\begin{align*}
bad & \leftrightarrow \frac{1}{3} \quad \rightarrow \quad abd & \leftrightarrow \frac{1}{3} \\
cea & \leftrightarrow \frac{2}{3} \quad \rightarrow \quad c\hat{a}e & \leftrightarrow \frac{2}{3}
\end{align*}
\]
Mazurkiewicz - Examples

\[ a \Join b, c, e; b \Join e; c \Join d; d \Join e \]

\[
\begin{align*}
bad & \leftrightarrow \frac{1}{3} \rightarrow abd & \leftrightarrow \frac{1}{3} \rightarrow abd & \rightarrow \frac{1}{3} \\
cea & \leftrightarrow \frac{2}{3} \rightarrow cae & \leftrightarrow \frac{2}{3} \rightarrow ace & \rightarrow \frac{2}{3}
\end{align*}
\]

The run obtained by the second scheduler
Confusion freeness

Theorem
In a probabilistic confusion free net, every two finite runs can be extended to Mazurkiewicz equivalent runs
Morally: the scheduling is irrelevant (up to fairness)
Petri nets: conclusions

What we have done

- definition of probabilistic Petri nets
- a semantics in terms of MDP
- a theorem on Mazurkiewicz equivalence

What we have left out

- Petri nets with symmetric confusion
- Infinite behaviour (fairness)
Road Map

- Interleaving
- Petri nets and Mazurkiewicz equivalence
  - Definitions
  - Confusion
  - Probabilistic Petri nets
  - Mazurkiewicz equivalence
- Event structures and tests
- Domains and valuations
Road Map

- Interleaving
- Petri nets and Mazurkiewicz equivalence
- Event structures and tests
- Domains and valuations
Road Map

- Interleaving
- Petri nets and Mazurkiewicz equivalence
- Event structures and tests
  - Definitions
  - Confusion
  - Probabilistic event structures
  - Tests
- Domains and valuations
Event Structures

Triple $\mathcal{E} = \langle E, \leq, \# \rangle$

$E$ is a set of events

$\leq$ is the casual dependency relation

$\#$ is the conflict relation

- $\langle E, \leq \rangle$ is a partial order
- for every $e \in E$, $e \downarrow$ is finite
- $\#$ is irreflexive and symmetric
- $\#$ is “hereditary”: $e_1 \# e$ and $e_1 \leq e_2$ implies $e_2 \# e$
Configurations

A notion of run

A configuration is a set $x$ of events

- justified: $e \in x, e' \leq e \implies e' \in x$
- conflict-free: $e, e' \in x \implies \neg e \# e'$

Examples:

$[e] := \{e' \mid e' \leq e\}$

$[e) := [e] \setminus \{e\}$

If $[e) \subseteq x$, $e$ is enabled at $x$

The set of configurations is $\mathcal{L}(\mathcal{E})$
Immediate Conflict

\[ e \#_{\mu} e' \iff \]
- \( e \# e' \) and
- \([e] \cup [e'], [e) \cup [e']\) are configurations

Two configurations \(x, x'\) are compatible if \(x \cup x'\) is a configuration
Petri Nets and Event structures

Event structures represent occurrence nets
Immediate conflict represents direct conflict
(Beware: immediate conflict in event structures does not coincide with direct conflict)
Event Structures: Examples

A configuration of $A$: $\{a, b, d\}$
A configuration of $B$: $\{a, b, d, e\}$
Probabilistic choice?

We want to resolve the conflicts by flipping a coin

How do we resolve the conflict between $b$, $c$, $d$?
Confusion

An event structure is confusion-free when

- (no symmetric confusion)
  \[ \#_\mu \cup 1_E \text{ is an equivalence} \]
- (no asymmetric confusion)
  \[ e \#_\mu e' \implies [e) = [e'] \]

The equivalence classes are traditionally called **cells**

A cell \( c \) is **enabled** at \( x \) if one (and therefore all) of its events is enabled at \( x \)

A cell \( c \) is **filled** by \( x \) if there is \( e \in c \cap x \)

A cell \( c \) is **accessible** at \( x \) if \( c \) is enabled at \( x \), but \( c \) is not filled by \( x \)
Confusion-freeness: Example

Configurations: \(\{a, b, f, g, q\}, \{a, b, e, n\}\)
Valuations on Event Structures

A valuation on $\mathcal{E}$ is a function $p : E \rightarrow ]0, 1]$ such that for every cell $c$

$$\sum_{e \in c} p(e) = 1$$

We define a function $v_p : \mathcal{L}(\mathcal{E}) \rightarrow [0, 1]$

$$v_p(x) = \prod_{e \in x} p(e)$$
Valuations: Example

Configurations:

\[ v_p(\{a, b, f, g, q\}) = \frac{1}{4}, \quad v_p(\{a, b, e, n\}) = \frac{1}{6} \]
Tests

A test is a set of configurations representing a probabilistic run

- $\{\emptyset\}$ is a test
- if $C$ is a test, $X \subseteq C$, for every $x \in X$, $c_x$ is a cell accessible at $x$, then

$$C \setminus \bigcup_{x \in X} \{x\} \cup \bigcup_{x \in X} \{x \cup e \mid e \in c_x\}$$
Properties of tests

When $C$ is a test:

- If $x, x' \in C$, then $x, x'$ are not compatible (incompatibility)
- Every configuration of $E$ is compatible with some $x \in C$ (maximality)

Runs can be extended:

$C \leq C'$ if for every $x \in C$ there is $x' \in C'$, $x \subseteq x'$
and for every $x' \in C'$ there is $x \in C$, $x \subseteq x'$
Tests: Example

Two tests

\{a, b, f, g, q\}, \{a, b, f, h, k\}, \{a, b, f, h, l\}

\{a, b, e, n\}, \{a, b, e, m\}, \{a, b, d, f\}, \{a, b, c, g\}, \{a, b, c, h\}
Tests: Example

\[ m_{\frac{1}{2}} \sim n_{\frac{1}{2}} \sim q_1 \]

\[ c_{\frac{1}{3}} \sim d_{\frac{1}{3}} \sim e_{\frac{1}{3}} \sim f_1 \sim g_{\frac{1}{4}} \sim h_{\frac{3}{4}} \sim k_{\frac{1}{5}} \sim l_{\frac{2}{5}} \]

\[ \emptyset \rightarrow \{a, b\} \]
Tests: Example

\{a, b\} \rightarrow \{a, b, c\}, \{a, b, d\}, \{a, b, e\}
Tests: Example

\[ m_{1/2} \sim n_{1/2} \quad q_1 \]

\[ c_{1/3} \sim d_{1/3} \sim e_{1/3} \quad f_1 \quad g_{1/4} \sim h_{3/4} \quad k_{1/5} \sim l_{2/5} \]

\[
\{a, b, c\} \rightarrow \{a, b, c, g\}, \{a, b, c, h\} \\
\{a, b, d\} \rightarrow \{a, b, d, f\} \\
\{a, b, e\} \rightarrow \{a, b, e, m\}, \{a, b, e, n\}
\]
Tests: Example

{a, b, e, n}, {a, b, e, m}, {a, b, d, f}, {a, b, c, g}, {a, b, c, h}
Tests: Example

\[ v_p(\{a, b, e, n\}) = \frac{1}{6}, \quad v_p(\{a, b, e, m\}) = \frac{1}{6} \]
\[ v_p(\{a, b, d, f\}) = \frac{1}{3}, \quad v_p(\{a, b, c, g\}) = \frac{1}{12} \]
\[ v_p(\{a, b, c, h\}) = \frac{1}{4} \]
Tests are runs

Proposition If $C$ is a test, then

$$\sum_{x \in C} v_p(x) = 1$$
Nondeterminism

There is a nondeterministic choice as to which cell to fire.

The two tests

\[ C := \{a, b, f, g, q\}, \{a, b, f, h, k\}, \{a, b, f, h, l\} \]
\[ C' := \{a, b, e, n\}, \{a, b, e, m\}, \{a, b, d, f\}, \{a, b, c, g\}, \{a, b, c, h\} \]

are incomparable

\[ C \not\preceq C', \quad C' \not\preceq C' \]

However
No Nondeterminism

Theorem

If $C, C'$ are tests, then there exists a test $C''$, with $C, C' \leq C''$

Morally: the nondeterministic branching does not matter
Conclusions

We have presented:

- A true concurrent model of probabilistic concurrency
- A notion of a run for such model
- A result about the behaviour of such runs
Road Map

- Interleaving
- Petri nets and Mazurkiewicz equivalence
- Event structures and tests
  - Definitions
  - Confusion
  - Probabilistic event structures
  - Tests
- Domains and valuations
Road Map

- Interleaving
- Petri nets and Mazurkiewicz equivalence
- Event structures and tests
- Domains and valuations
Road Map

- Interleaving
- Petri nets and Mazurkiewicz equivalence
- Event structures and tests
- Domains and valuations
  - Domain of configurations
  - Continuous valuations
  - Probabilistic event structures and valuations
  - Beyond stochastic independence
Event Structures and Domains

The set $\mathcal{L}(\mathcal{E})$ of configurations ordered by inclusion

- is an algebraic DCPO
- its compact elements are the finite configurations

If the event structure is confusion-free, $\mathcal{L}(\mathcal{E})$ is a concrete domain (Kahn and Plotkin)
Continuous Valuations

A continuous valuation on a topological space \((X, \tau)\) is a function \(\nu : \tau \rightarrow \mathbb{R}^+\) satisfying:

- (Strictness) \(\nu(\emptyset) = 0\)
- (Monotonicity) \(U \subseteq V \implies \nu(U) \leq \nu(V)\)
- (Modularity) \(\nu(U) + \nu(V) = \nu(U \cup V) + \nu(U \cap V)\)
- (Continuity) whenever \(\mathcal{J}\) is a directed subset of \(\tau\)

\[\nu\left(\bigcup \mathcal{J}\right) = \sup_{U \in \mathcal{J}} \nu(U)\]
Valuations on Domains

A subset $O$ of a DCPO is Scott-open if

- $O$ is upward closed
- $O$ is “inaccessible”: if $X$ is directed and $X \cap O = \emptyset$ then $\bigcup \uparrow X \not\in O$

$d$ is compact, if and only if $d \uparrow$ is open

A continuous valuation on a DCPO is a continuous valuation on its Scott topology
Valuations and Measures

A continuous valuation on an algebraic domain extends to Borel measure. A Borel measure on a $\omega$-algebraic domain restricts to a continuous valuation.
Event Structures and Domains

Theorem

For every valuation $p$ on an event structure $\mathcal{E}$ there is a unique continuous valuation $\nu_p$ on $\mathcal{L}(\mathcal{E})$ such that, for every finite configuration $x$:

$$\nu_p(x \uparrow) = \nu_p(x) = \prod_{e \in x} p(x)$$
Valuations and independence

Not all continuous valuations are obtained in this way

\[ a \sim b \quad c \sim d \]

This is \( \mathcal{L}(\mathcal{E}) \):

\[
\begin{align*}
\{a, c\} & \quad \{a, d\} & \quad \{b, c\} & \quad \{b, d\} \\
\{a\} & \quad \{b\} & \quad \{c\} & \quad \{d\} \\
\emptyset & & & \end{align*}
\]
Valuations and independence

Now put

- $\xi(\emptyset \uparrow) = 1$
- $\xi(\{a\} \uparrow) = \xi(\{b\} \uparrow) = \xi(\{c\} \uparrow) = \xi(\{d\} \uparrow) = 1/2$
- $\xi(\{a, c\} \uparrow) = \xi(\{b, d\} \uparrow) = 0$
- $\xi(\{a, d\} \uparrow) = \xi(\{b, c\} \uparrow) = 1/2$

No valuation on $E$ generates $\xi$

There is a correlation between the two cells
Without independence

A **valuation** on $\mathcal{E}$ is a function $\nu : \mathcal{L}_{fin}(E) \rightarrow [0, 1]$ such that for every test $C$

$$\sum_{x \in C} \nu(x) = 1$$

**Theorem**

For every valuation $\nu$ on an event structure $\mathcal{E}$ there is a unique continuous valuation $\nu_v$ on $\mathcal{L}(\mathcal{E})$ such that, for every finite configuration $x$:

$$\nu_v(x \uparrow) = \nu(x)$$
Morphisms

A morphism $f : \mathcal{E} \to \mathcal{E}'$ is a (partial) function from $E \to E'$ such that if $x$ is a configuration, $f(x)$ is a configuration.
The role of morphisms

How we get valuations without independence

\[ b' \sim c' \quad d' \sim e' \quad b'' \sim c'' \quad d'' \sim e'' \]

\[ a' \sim \cdots \sim a'' \]

\[ b \sim c \quad d \sim e \]

\[ a \]
The role of morphisms

How we get valuations without independence

A morphism $f : \mathcal{E} \rightarrow \mathcal{E}'$
The role of morphisms

How we get valuations without independence

\[ b' \sim c' \quad d' \sim e' \quad b'' \sim c'' \quad d'' \sim e'' \]

\[ a' \sim \sim \sim \sim \sim \sim a'' \]

\[ b \sim c \quad d \sim e \]

\[ a \]

A morphism \( f : \mathcal{E} \rightarrow \mathcal{E}' \)
The role of morphisms

How we get valuations without independence

A morphism $f : \mathcal{E} \rightarrow \mathcal{E}'$
The role of morphisms

How we get valuations without independence

A morphism \( f : \mathcal{E} \to \mathcal{E'} \)
The role of morphisms

How we get valuations without independence

\[ b' \sim c' \quad d' \sim e' \quad b'' \sim c'' \quad d'' \sim e'' \]

\[ a'_1 \sim a''_1 \]

\[ b \sim c \quad d \sim e \]

A valuation \( \nu \) on \( \mathcal{E} \)
The role of morphisms

How we get valuations without independence

\[ b_1' \sim c_0' \quad d_1' \sim e_0' \quad b_0'' \sim c_1'' \quad d_0'' \sim e_1'' \]

\[ a_{1/2}' \sim \sim \sim \sim \sim \sim a_{1/2}'' \]

\[ b \sim c \quad d \sim e \]

A valuation \( \nu \) on \( \mathcal{E} \)
The role of morphisms

How we get valuations without independence

\[ b'_1 \sim c'_0 \quad d'_1 \sim e'_0 \quad b''_0 \sim c''_1 \quad d''_0 \sim e''_1 \]

\[ a'_{\frac{1}{2}} \sim \cdots \sim \sim a''_{\frac{1}{2}} \]

\[ b \sim c \quad d \sim e \]

Define a valuation \( \nu' \) on \( \mathcal{E}' \) by pulling back \( \nu \)
The role of morphisms

How we get valuations without independence

\[ b'_1 \sim c'_0 \quad d'_1 \sim e'_0 \quad b''_0 \sim c''_1 \quad d''_0 \sim e''_1 \]

\[ a'_{\frac{1}{2}} \sim \cdots \sim a''_{\frac{1}{2}} \]

\[ b \sim c \quad d \sim e \]

\[ \nu'(y) = \sum_{f(x) = y} \nu(x) \]
The role of morphisms

How we get valuations without independence

\[ b'_1 \sim c'_0 \quad d'_1 \sim e'_0 \quad b''_0 \sim c''_1 \quad d''_0 \sim e''_1 \]

\[ a'_{\frac{1}{2}} \sim \sim \sim \sim \sim \sim \sim \sim a''_{\frac{1}{2}} \]

\[ b \sim c \quad d \sim e \]

\[ \nu'(\{a, b\}) = \frac{1}{2}, \nu'(\{a, e\}) = \frac{1}{2} \]

\[ \nu'(\{a, b, e\}) = 0 \]
The role of morphisms

How we get valuations without independence

\[ b'_1 \sim c'_0 \quad d'_1 \sim e'_0 \quad b''_0 \sim c''_1 \quad d''_0 \sim e''_1 \]

\[ a'_{\frac{1}{2}} \sim \ldots \sim a''_{\frac{1}{2}} \]

\[ b \sim c \quad d \sim e \]

Negative correlation between \( b \) and \( e \)
Due to a hidden choice
Conclusions

We have presented

- Relations with domain theory
- How to go beyond independence
Related work

- Katoen’s Probabilistic Event Structures
- Völzer’s thesis.
- Benveniste, Fabre, Haar: Markov Nets.
Future (present?) work

- relating the related work
- beyond confusion freeness
- “concrete” applications
- continuous probabilities
- bisimulation, logics, verification...