Labelled Markov Processes

Lecture 2: Probabilistic Transition Systems

Prakash Panangaden[1]

[1]School of Computer Science
McGill University

January 2008, Winter School on Logic, IIT Kanpur

Outline

Outline

Outline

Outline

Outline

Overview

- Discrete probabilistic transition system.
- Labelled Markov processes: probabilistic transition systems with continuous state spaces.
- Bisimulation for LMPs.
- Logical characterization.
- Simulation.

Overview

- Discrete probabilistic transition system.
- Labelled Markov processes: probabilistic transition systems with continuous state spaces.
- Bisimulation for LMPs.
- Logical characterization.
- Simulation.

Overview

- Discrete probabilistic transition system.
- Labelled Markov processes: probabilistic transition systems with continuous state spaces.
- Bisimulation for LMPs.
- Logical characterization.
- Simulation.

Overview

- Discrete probabilistic transition system.
- Labelled Markov processes: probabilistic transition systems with continuous state spaces.
- Bisimulation for LMPs.
- Logical characterization.
- Simulation.

Overview

- Discrete probabilistic transition system.
- Labelled Markov processes: probabilistic transition systems with continuous state spaces.
- Bisimulation for LMPs.
- Logical characterization.
- Simulation.

Summary of Results

- Probabilistic bisimulation can be defined for continuous state-space systems. [LICS97]
- Logical characterization. [LICS98,Info and Comp 2002]
- Metric analogue of bisimulation. [CONCUR99, TCS2004]
- Approximation of LMPs. [LICS00,Info and Comp 2003]
- Weak bisimulation. [LICS02,CONCUR02]
- Real time. [QEST 2004, JLAP 2003,LMCS 2006]

Summary of Results

- Probabilistic bisimulation can be defined for continuous state-space systems. [LICS97]
- Logical characterization. [LICS98,Info and Comp 2002]
- Metric analogue of bisimulation. [CONCUR99, TCS2004]
- Approximation of LMPs. [LICS00,Info and Comp 2003]
- Weak bisimulation. [LICS02,CONCUR02]
- Real time. [QEST 2004, JLAP 2003,LMCS 2006]

Summary of Results

- Probabilistic bisimulation can be defined for continuous state-space systems. [LICS97]
- Logical characterization. [LICS98,Info and Comp 2002]
- Metric analogue of bisimulation. [CONCUR99, TCS2004]
- Approximation of LMPs. [LICS00,Info and Comp 2003]
- Weak bisimulation. [LICS02,CONCUR02]
- Real time. [QEST 2004, JLAP 2003,LMCS 2006]

Summary of Results

- Probabilistic bisimulation can be defined for continuous state-space systems. [LICS97]
- Logical characterization. [LICS98,Info and Comp 2002]
- Metric analogue of bisimulation. [CONCUR99, TCS2004]
- Approximation of LMPs. [LICS00,Info and Comp 2003]
- Weak bisimulation. [LICS02,CONCUR02]
- Real time. [QEST 2004, JLAP 2003,LMCS 2006]

Summary of Results

- Probabilistic bisimulation can be defined for continuous state-space systems. [LICS97]
- Logical characterization. [LICS98,Info and Comp 2002]
- Metric analogue of bisimulation. [CONCUR99, TCS2004]
- Approximation of LMPs. [LICS00,Info and Comp 2003]
- Weak bisimulation. [LICS02,CONCUR02]
- Real time. [QEST 2004, JLAP 2003,LMCS 2006]

Summary of Results

- Probabilistic bisimulation can be defined for continuous state-space systems. [LICS97]
- Logical characterization. [LICS98,Info and Comp 2002]
- Metric analogue of bisimulation. [CONCUR99, TCS2004]
- Approximation of LMPs. [LICS00,Info and Comp 2003]
- Weak bisimulation. [LICS02,CONCUR02]
- Real time. [QEST 2004, JLAP 2003,LMCS 2006]

Collaborators

- Josée Desharnais
- Radha Jagadeesan and Vineet Gupta
- Abbas Edalat
- Vincent Danos

Collaborators

- Josée Desharnais
- Radha Jagadeesan and Vineet Gupta
- Abbas Edalat
- Vincent Danos

Collaborators

- Josée Desharnais
- Radha Jagadeesan and Vineet Gupta
- Abbas Edalat
- Vincent Danos

Collaborators

- Josée Desharnais
- Radha Jagadeesan and Vineet Gupta
- Abbas Edalat
- Vincent Danos

Labelled Transition System

- A set of states S,
- a set of *labels* or *actions*, L or $\mathcal{A}$ and
- a transition relation $\subseteq S \times \mathcal{A} \times S$, usually written

$$\rightarrow_a \subseteq S \times S.$$

The transitions could be indeterminate (nondeterministic).

Labelled Transition System

- A set of states S,
- a set of *labels* or *actions*, L or $\mathcal{A}$ and
- a transition relation $\subseteq S \times \mathcal{A} \times S$, usually written

$$\rightarrow_a \subseteq S \times S.$$

The transitions could be indeterminate (nondeterministic).

Labelled Transition System

- A set of states S,
- a set of *labels* or *actions*, L or $\mathcal{A}$ and
- a transition relation $\subseteq S \times \mathcal{A} \times S$, usually written

$$\to_a \subseteq S \times S.$$

The transitions could be indeterminate (nondeterministic).

Markov Chains

- A *discrete-time* Markov chain is a finite set S (the state space) together with a transition probability function $T : S \times S \to [0, 1]$.
- A Markov chain is just a probabilistic automaton; if we add labels we get a PTS.
- The key property is that the transition probability from s to s' only depends on s and s' and not on the past history of how it got there. This is what allows the probabilistic data to be given as a single matrix T.

Markov Chains

- A *discrete-time* Markov chain is a finite set S (the state space) together with a transition probability function $T : S \times S \longrightarrow [0, 1]$.
- A Markov chain is just a probabilistic automaton; if we add labels we get a PTS.
- The key property is that the transition probability from s to s' only depends on s and s' and not on the past history of how it got there. This is what allows the probabilistic data to be given as a single matrix T.

Markov Chains

- A *discrete-time* Markov chain is a finite set S (the state space) together with a transition probability function $T : S \times S \longrightarrow [0, 1]$.
- A Markov chain is just a probabilistic automaton; if we add labels we get a PTS.
- The key property is that the transition probability from s to s' only depends on s and s' and not on the past history of how it got there. This is what allows the probabilistic data to be given as a single matrix T.

Discrete probabilistic transition systems

- Just like a labelled transition system with probabilities associated with the transitions.
- $$(S, \mathrm{L}, \forall a \in \mathrm{L}\ T_a : S \times S \longrightarrow [0, 1])$$
- The model is *reactive*: All probabilistic data is *internal* - no probabilities associated with environment behaviour.

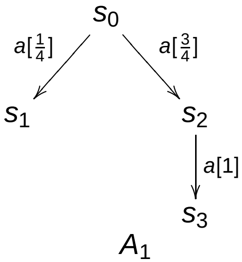
Discrete probabilistic transition systems

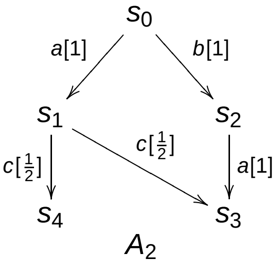
- Just like a labelled transition system with probabilities associated with the transitions.
- $$(S, \mathrm{L}, \forall a \in \mathrm{L}\ T_a : S \times S \longrightarrow [0, 1])$$
- The model is *reactive*: All probabilistic data is *internal* - no probabilities associated with environment behaviour.

Discrete probabilistic transition systems

- Just like a labelled transition system with probabilities associated with the transitions.
- $$(S, \mathrm{L}, \forall a \in \mathrm{L}\ T_a : S \times S \longrightarrow [0,1])$$
- The model is *reactive*: All probabilistic data is *internal* - no probabilities associated with environment behaviour.

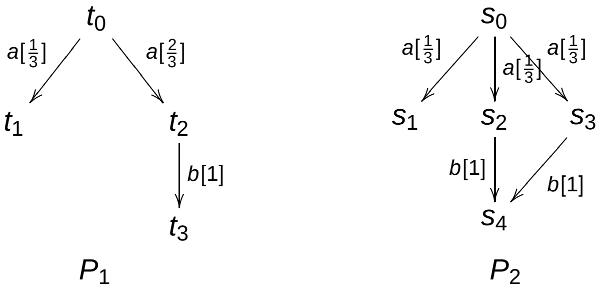
Examples of PTSs





Bisimulation for PTS: Larsen and Skou

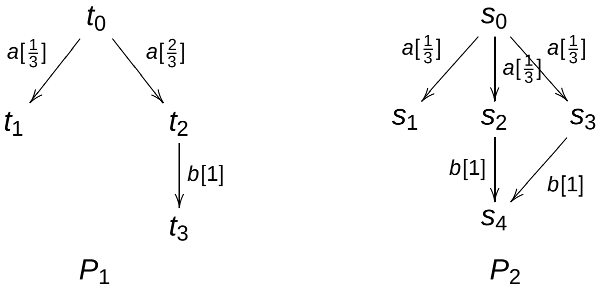
- Consider



- Should s_0 and t_0 be bisimilar?
- Yes, but we need to add the probabilities.

Bisimulation for PTS: Larsen and Skou

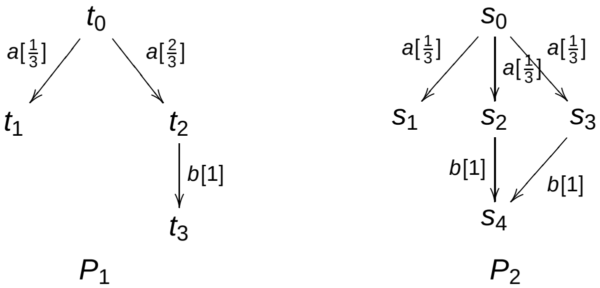
- Consider



- Should s_0 and t_0 be bisimilar?
- Yes, but we need to add the probabilities.

Bisimulation for PTS: Larsen and Skou

- Consider



- Should s_0 and t_0 be bisimilar?
- Yes, but we need to add the probabilities.

The Official Definition

- Let $\mathcal{S} = (S, \mathrm{L}, T_a)$ be a PTS. An equivalence relation R on S is a **bisimulation** if whenever sRs', with $s, s' \in S$, we have that for all $a \in \mathcal{A}$ and every R-equivalence class, A, $T_a(s, A) = T_a(s', A)$.
- The notation $T_a(s, A)$ means "the probability of starting from s and jumping to a state in the set A."
- Two states are bisimilar if there is some bisimulation relation R relating them.

The Official Definition

- Let $\mathcal{S} = (S, \mathsf{L}, T_a)$ be a PTS. An equivalence relation R on S is a **bisimulation** if whenever sRs', with $s, s' \in S$, we have that for all $a \in \mathcal{A}$ and every R-equivalence class, A, $T_a(s, A) = T_a(s', A)$.
- The notation $T_a(s, A)$ means "the probability of starting from s and jumping to a state in the set A."
- Two states are bisimilar if there is some bisimulation relation R relating them.

The Official Definition

- Let $\mathcal{S} = (S, \mathrm{L}, T_a)$ be a PTS. An equivalence relation R on S is a **bisimulation** if whenever sRs', with $s, s' \in S$, we have that for all $a \in \mathcal{A}$ and every R-equivalence class, A, $T_a(s, A) = T_a(s', A)$.
- The notation $T_a(s, A)$ means "the probability of starting from s and jumping to a state in the set A."
- Two states are bisimilar if there is some bisimulation relation R relating them.

What are labelled Markov processes?

- Labelled Markov processes are probabilistic versions of labelled transition systems. Labelled transition systems where the final state is governed by a probability distribution - no other indeterminacy.
- All probabilistic data is *internal* - no probabilities associated with environment behaviour.
- We observe the interactions - not the internal states.
- **In general, the state space of a labelled Markov process may be a *continuum*.**

What are labelled Markov processes?

- Labelled Markov processes are probabilistic versions of labelled transition systems. Labelled transition systems where the final state is governed by a probability distribution - no other indeterminacy.
- All probabilistic data is *internal* - no probabilities associated with environment behaviour.
- We observe the interactions - not the internal states.
- **In general, the state space of a labelled Markov process may be a *continuum*.**

What are labelled Markov processes?

- Labelled Markov processes are probabilistic versions of labelled transition systems. Labelled transition systems where the final state is governed by a probability distribution - no other indeterminacy.
- All probabilistic data is *internal* - no probabilities associated with environment behaviour.
- We observe the interactions - not the internal states.
- In general, the state space of a labelled Markov process may be a *continuum*.

What are labelled Markov processes?

- Labelled Markov processes are probabilistic versions of labelled transition systems. Labelled transition systems where the final state is governed by a probability distribution - no other indeterminacy.
- All probabilistic data is *internal* - no probabilities associated with environment behaviour.
- We observe the interactions - not the internal states.
- **In general, the state space of a labelled Markov process may be a *continuum*.**

Motivation

Model and reason about systems with *continuous* state spaces or continuous time evolution or both.

- hybrid control systems; e.g. flight management systems.
- telecommunication systems with spatial variation; e.g. cell phones
- performance modelling,
- continuous time systems,
- probabilistic process algebra with recursion.

Motivation

Model and reason about systems with *continuous* state spaces or continuous time evolution or both.

- hybrid control systems; e.g. flight management systems.
- telecommunication systems with spatial variation; e.g. cell phones
- performance modelling,
- continuous time systems,
- probabilistic process algebra with recursion.

Motivation

Model and reason about systems with *continuous* state spaces or continuous time evolution or both.

- hybrid control systems; e.g. flight management systems.
- telecommunication systems with spatial variation; e.g. cell phones
- performance modelling,
- continuous time systems,
- probabilistic process algebra with recursion.

Motivation

Model and reason about systems with *continuous* state spaces or continuous time evolution or both.

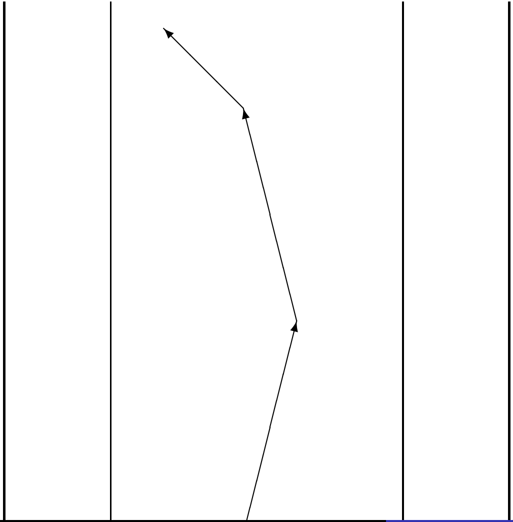
- hybrid control systems; e.g. flight management systems.
- telecommunication systems with spatial variation; e.g. cell phones
- performance modelling,
- continuous time systems,
- probabilistic process algebra with recursion.

Motivation

Model and reason about systems with *continuous* state spaces or continuous time evolution or both.

- hybrid control systems; e.g. flight management systems.
- telecommunication systems with spatial variation; e.g. cell phones
- performance modelling,
- continuous time systems,
- probabilistic process algebra with recursion.

An Example of a Continuous-State System



a - turn left

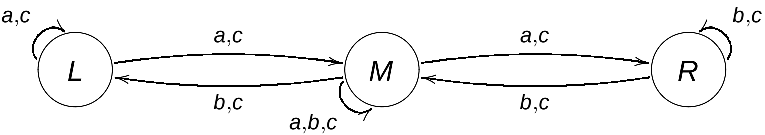
b - turn right

c - straight

Actions

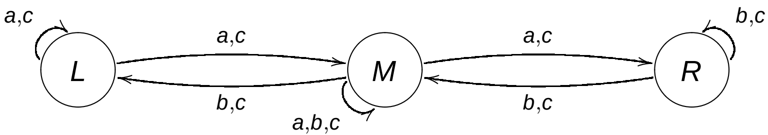
a - turn left, *b* - turn right, *c* - keep on course
The actions move the craft sideways with some probability distributions on how far it moves. The craft may “drift” even with *c*. The action *a* (*b*) must be disabled when the craft is too near the left (right) boundary.

Schematic of Example

- 

- This picture is misleading: unless very special conditions hold the process cannot be compressed into an *equivalent* (?) finite-state model. In general, the transition probabilities should depend on the position.

Schematic of Example

- 

- This picture is misleading: unless very special conditions hold the process cannot be compressed into an *equivalent* (?) finite-state model. In general, the transition probabilities should depend on the position.

Some remarks on the use of this model

- This is a toy model but exemplifies the issues.
- Can be used for reasoning - much better if we could have a finite-state version.
- Why not discretize right away and never worry about the continuous case? Because we lose the ability to *refine* the model later.
- A better model would be to base it on rewards and think about finiding optimal policies as in AI literature.

Some remarks on the use of this model

- This is a toy model but exemplifies the issues.
- Can be used for reasoning - much better if we could have a finite-state version.
- Why not discretize right away and never worry about the continuous case? Because we lose the ability to *refine* the model later.
- A better model would be to base it on rewards and think about finiding optimal policies as in AI literature.

Some remarks on the use of this model

- This is a toy model but exemplifies the issues.
- Can be used for reasoning - much better if we could have a finite-state version.
- Why not discretize right away and never worry about the continuous case? Because we lose the ability to *refine* the model later.
- A better model would be to base it on rewards and think about finiding optimal policies as in AI literature.

Some remarks on the use of this model

- This is a toy model but exemplifies the issues.
- Can be used for reasoning - much better if we could have a finite-state version.
- Why not discretize right away and never worry about the continuous case? Because we lose the ability to *refine* the model later.
- A better model would be to base it on rewards and think about finiding optimal policies as in AI literature.

The Need for Measure Theory

- Basic fact: There are subsets of **R** for which no sensible notion of size can be defined.
- More precisely, there is no translation-invariant measure defined on all the subsets of the reals.

The Need for Measure Theory

- Basic fact: There are subsets of **R** for which no sensible notion of size can be defined.
- More precisely, there is no translation-invariant measure defined on all the subsets of the reals.

Stochastic Kernels

- A *stochastic kernel* (Markov kernel) is a function $h : S \times \Sigma \to [0,1]$ with (a) $h(s,\cdot) : \Sigma \to [0,1]$ a (sub)probability measure and (b) $h(\cdot, A) : X \to [0,1]$ a measurable function.
- Though apparantly asymmetric, these are the stochastic analogues of binary relations
- and the uncountable generalization of a matrix.

Stochastic Kernels

- A *stochastic kernel* (Markov kernel) is a function $h : S \times \Sigma \to [0, 1]$ with (a) $h(s, \cdot) : \Sigma \to [0, 1]$ a (sub)probability measure and (b) $h(\cdot, A) : X \to [0, 1]$ a measurable function.
- Though apparently asymmetric, these are the stochastic analogues of binary relations
- and the uncountable generalization of a matrix.

Stochastic Kernels

- A *stochastic kernel* (Markov kernel) is a function $h : S \times \Sigma \to [0, 1]$ with (a) $h(s, \cdot) : \Sigma \to [0, 1]$ a (sub)probability measure and (b) $h(\cdot, A) : X \to [0, 1]$ a measurable function.
- Though apparently asymmetric, these are the stochastic analogues of binary relations
- and the uncountable generalization of a matrix.

Formal Definition of LMPs

- An LMP is a tuple $(S, \Sigma, \mathsf{L}, \forall \alpha \in \mathsf{L}.\tau_\alpha)$ where $\tau_\alpha : S \times \Sigma \to [0,1]$ is a *transition probability* function such that
- $\forall s : S.\lambda A : \Sigma.\tau_\alpha(s, A)$ is a subprobability measure and $\forall A : \Sigma.\lambda s : S.\tau_\alpha(s, A)$ is a measurable function.

Formal Definition of LMPs

- An LMP is a tuple $(S, \Sigma, \mathsf{L}, \forall\alpha \in \mathsf{L}.\tau_\alpha)$ where $\tau_\alpha : S \times \Sigma \to [0,1]$ is a *transition probability* function such that
- $\forall s : S.\lambda A : \Sigma.\tau_\alpha(s, A)$ is a subprobability measure and
  $\forall A : \Sigma.\lambda s : S.\tau_\alpha(s, A)$ is a measurable function.

Larsen-Skou Bisimulation

- Let $\mathcal{S} = (S, i, \Sigma, \tau)$ be a labelled Markov process. An equivalence relation R on S is a **bisimulation** if whenever sRs', with $s, s' \in S$, we have that for all $a \in \mathcal{A}$ and every R-closed measurable set $A \in \Sigma$, $\tau_a(s, A) = \tau_a(s', A)$. Two states are bisimilar if they are related by a bisimulation relation.
- Can be extended to bisimulation between two different **LMP**s.

Larsen-Skou Bisimulation

- Let $\mathcal{S} = (S, i, \Sigma, \tau)$ be a labelled Markov process. An equivalence relation R on S is a **bisimulation** if whenever sRs', with $s, s' \in S$, we have that for all $a \in \mathcal{A}$ and every R-closed measurable set $A \in \Sigma$, $\tau_a(s, A) = \tau_a(s', A)$.
  Two states are bisimilar if they are related by a bisimulation relation.
- Can be extended to bisimulation between two different **LMP**s.

Logical Characterization

- $$\mathcal{L} ::== \mathsf{T} | \phi_1 \wedge \phi_2 | \langle a \rangle_q \phi$$
- We say $s \models \langle a \rangle_q \phi$ iff
$$\exists A \in \Sigma.(\forall s' \in A. s' \models \phi) \wedge (\tau_a(s, A) > q).$$
- Two systems are bisimilar iff they obey the same formulas of $\mathcal{L}$. [DEP 1998 LICS, I and C 2002]

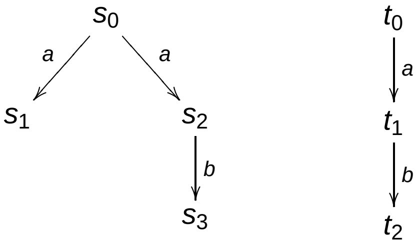
Logical Characterization

- $$\mathcal{L} ::== \mathsf{T} | \phi_1 \wedge \phi_2 | \langle a \rangle_q \phi$$
- We say $s \models \langle a \rangle_q \phi$ iff
$$\exists A \in \Sigma. (\forall s' \in A. s' \models \phi) \wedge (\tau_a(s, A) > q).$$
- Two systems are bisimilar iff they obey the same formulas of $\mathcal{L}$. [DEP 1998 LICS, I and C 2002]

Logical Characterization

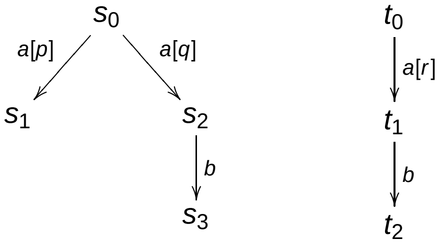
- $$\mathcal{L} ::== \mathsf{T} | \phi_1 \wedge \phi_2 | \langle a \rangle_q \phi$$
- We say $s \models \langle a \rangle_q \phi$ iff
  $$\exists A \in \Sigma . (\forall s' \in A . s' \models \phi) \wedge (\tau_a(s, A) > q).$$
- Two systems are bisimilar iff they obey the same formulas of $\mathcal{L}$. [DEP 1998 LICS, I and C 2002]

That cannot be right?



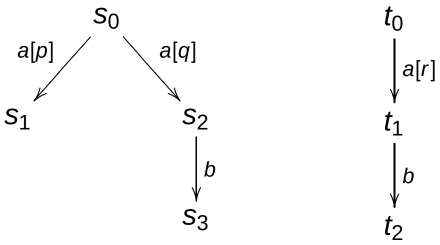
Two processes that cannot be distinguished without negation.
The formula that distinguishes them is $\langle a \rangle (\neg \langle b \rangle \top)$.

But it is!

- 

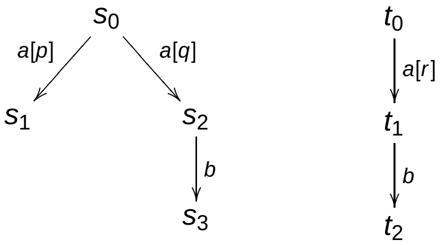
  We add probabilities to the transitions.
- If $p + q < r$ or $p + q > r$ we can easily distinguish them.
- If $p + q = r$ and $p > 0$ then $q < r$ so $\langle a \rangle_r \langle b \rangle_1 \top$ distinguishes them.

But it is!

- 

  We add probabilities to the transitions.
- If $p + q < r$ or $p + q > r$ we can easily distinguish them.
- If $p + q = r$ and $p > 0$ then $q < r$ so $\langle a \rangle_r \langle b \rangle_1 \top$ distinguishes them.

But it is!

- 

  We add probabilities to the transitions.
- If $p + q < r$ or $p + q > r$ we can easily distinguish them.
- If $p + q = r$ and $p > 0$ then $q < r$ so $\langle a \rangle_r \langle b \rangle_1 \top$ distinguishes them.

Proof idea

- Show that the relation "s and s' satisfy exactly the same formulas" is a bisimulation.
- Can easily show that $\tau_a(s, A) = \tau_a(s', A)$ for A of the form $[\![\phi]\!]$.
- Use Dynkin's lemma to show that we get a well defined measure on the σ-algebra generated by such sets and the above equality holds.
- Use special properties of analytic spaces to show that this σ-algebra is the same as the original σ-algebra.

Proof idea

- Show that the relation "s and s' satisfy exactly the same formulas" is a bisimulation.
- Can easily show that $\tau_a(s, A) = \tau_a(s', A)$ for A of the form $[\![\phi]\!]$.
- Use Dynkin's lemma to show that we get a well defined measure on the σ-algebra generated by such sets and the above equality holds.
- Use special properties of analytic spaces to show that this σ-algebra is the same as the original σ-algebra.

Proof idea

- Show that the relation "s and s' satisfy exactly the same formulas" is a bisimulation.
- Can easily show that $\tau_a(s, A) = \tau_a(s', A)$ for A of the form $[\![\phi]\!]$.
- Use Dynkin's lemma to show that we get a well defined measure on the σ-algebra generated by such sets and the above equality holds.
- Use special properties of analytic spaces to show that this σ-algebra is the same as the original σ-algebra.

Proof idea

- Show that the relation "s and s' satisfy exactly the same formulas" is a bisimulation.
- Can easily show that $\tau_a(s, A) = \tau_a(s', A)$ for A of the form $[\![\phi]\!]$.
- Use Dynkin's lemma to show that we get a well defined measure on the σ-algebra generated by such sets and the above equality holds.
- Use special properties of analytic spaces to show that this σ-algebra is the same as the original σ-algebra.

Simulation

Let $\mathcal{S} = (S, \Sigma, \tau)$ be a labelled Markov process. A preorder R on S is a **simulation** if whenever sRs', we have that for all $a \in \mathcal{A}$ and every R-closed measurable set $A \in \Sigma$, $\tau_a(s, A) \leq \tau_a(s', A)$. We say s is simulated by s' if sRs' for some simulation relation R.

Logic for simulation?

- The logic used in the characterization has no negation, not even a limited negative construct.
- One can show that if s simulates s' then s satisfies all the formulas of $\mathcal{L}$ that s' satisfies.
- What about the converse?

Logic for simulation?

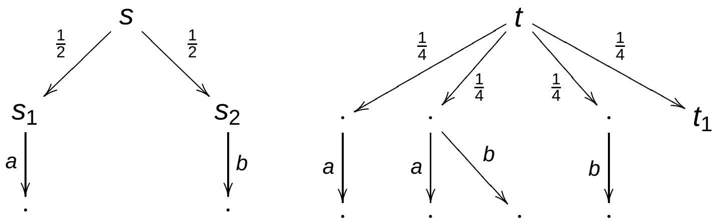
- The logic used in the characterization has no negation, not even a limited negative construct.
- One can show that if s simulates s' then s satisfies all the formulas of $\mathcal{L}$ that s' satisfies.
- What about the converse?

Logic for simulation?

- The logic used in the characterization has no negation, not even a limited negative construct.
- One can show that if s simulates s' then s satisfies all the formulas of $\mathcal{L}$ that s' satisfies.
- What about the converse?

Counter example!

In the following picture, t satisfies all formulas of $\mathcal{L}$ that s satisfies but t does not simulate s.



All transitions from s and t are labelled by a.

Counter example (contd.)

- A formula of $\mathcal{L}$ that is satisfied by t but not by s.

$$\langle a\rangle_0(\langle a\rangle_0\mathrm{T} \wedge \langle b\rangle_0\mathrm{T}).$$

- A formula with disjunction that is satisfied by s but not by t:

$$\langle a\rangle_{\frac{3}{4}}(\langle a\rangle_0\mathrm{T} \vee \langle b\rangle_0\mathrm{T}).$$

Counter example (contd.)

- A formula of $\mathcal{L}$ that is satisfied by t but not by s.

$$\langle a\rangle_0(\langle a\rangle_0 \mathrm{T} \wedge \langle b\rangle_0 \mathrm{T}).$$

- A formula with disjunction that is satisfied by s but not by t:

$$\langle a\rangle_{\frac{3}{4}}(\langle a\rangle_0 \mathrm{T} \vee \langle b\rangle_0 \mathrm{T}).$$

A logical characterization for simulation

- The logic $\mathcal{L}$ does **not** characterize simulation. One needs disjunction.

$$\mathcal{L}_\vee := \mathcal{L}\phi 1 \vee \phi_2.$$

- With this logic we have:
An **LMP** s_1 simulates s_2 if and only if for every formula ϕ of $\mathcal{L}_\vee$ we have

$$s_1 \models \phi \Rightarrow s_2 \models \phi.$$

- The only proof we know uses domain theory.

A logical characterization for simulation

- The logic $\mathcal{L}$ does **not** characterize simulation. One needs disjunction.

$$\mathcal{L}_\vee := \mathcal{L}\phi 1 \vee \phi_2.$$

- With this logic we have:
  An **LMP** s_1 simulates s_2 if and only if for every formula ϕ of $\mathcal{L}_\vee$ we have

$$s_1 \models \phi \Rightarrow s_2 \models \phi.$$

- The only proof we know uses domain theory.

A logical characterization for simulation

- The logic $\mathcal{L}$ does **not** characterize simulation. One needs disjunction.

$$\mathcal{L}_\vee := \mathcal{L}\phi 1 \vee \phi_2.$$

- With this logic we have:
  An **LMP** s_1 simulates s_2 if and only if for every formula ϕ of $\mathcal{L}_\vee$ we have
  $$s_1 \models \phi \Rightarrow s_2 \models \phi.$$
- The only proof we know uses domain theory.