Simulation and Bisimulation for Probabilistic Timed Automata

Algorithms and Logical Characterization

Jeremy Sproston and Angelo Troina

Dipartimento di Informatica
University of Turin
Italy

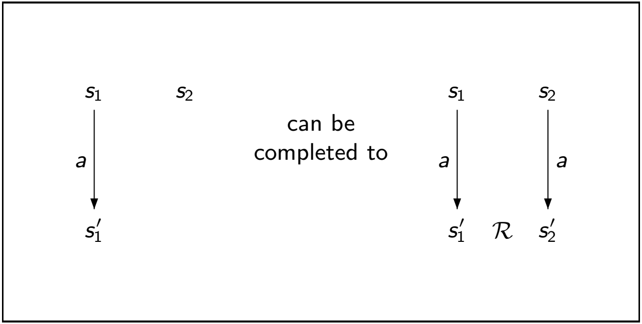
PaCo @ ICTCS 2010
15th September 2010

Motivation

- Aim: construction of abstractions (or refinements) of probabilistic timed automata.
- Extensive work on abstraction for probabilistic labelled transition systems and timed automata, often based (in part) on simulation or bisimulation relations.
- Method: combine techniques from probabilistic labelled transition systems and timed automata to use (bi)simulation for probabilistic timed automata.
- In particular, study algorithms and logical characterization.

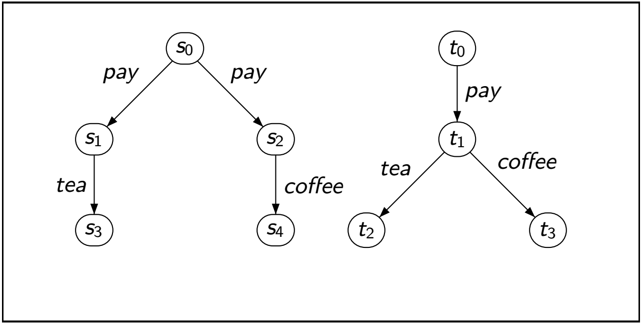
Simulation

- Labelled transition system $(S, Act, \rightarrow)$, where $\rightarrow \subseteq S \times Act \times S$ (write $s \xrightarrow{a} s'$ to denote $(s, a, s') \in \rightarrow$).
- Relation $\mathcal{R} \subseteq S \times S$ is a simulation relation if $\mathcal{R}$ satisfies the following condition:
  $(s_1, s_2) \in \mathcal{R}$ implies that, for each $s_1 \xrightarrow{a} s_1'$, there exists $s_2 \xrightarrow{a} s_2'$ such that $(s_1', s_2') \in \mathcal{R}$.



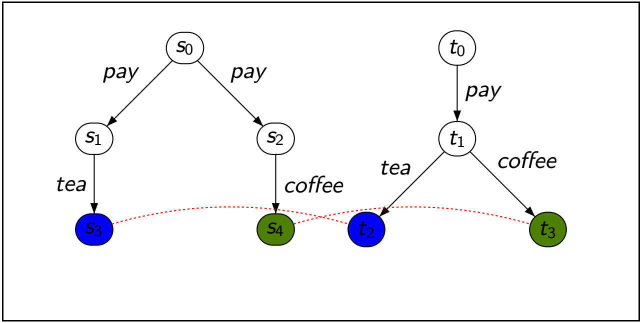
Simulation

- Example: does t_0 simulate s_0, given that t_2 simulates s_3, and t_3 simulates s_4?



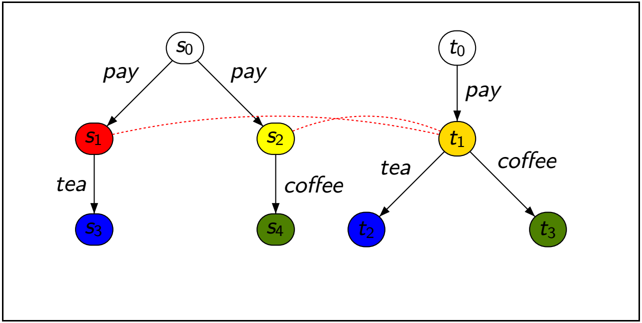
Simulation

- Example:



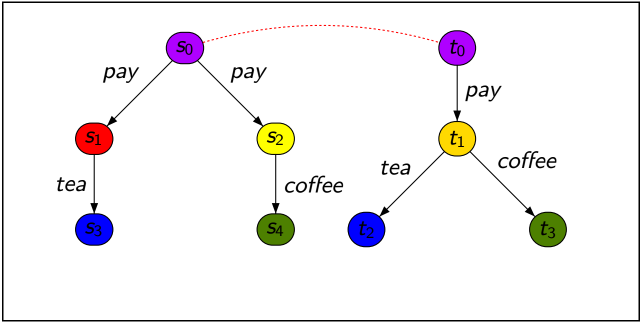
Simulation

- Example: t_1 simulates s_1 and s_2



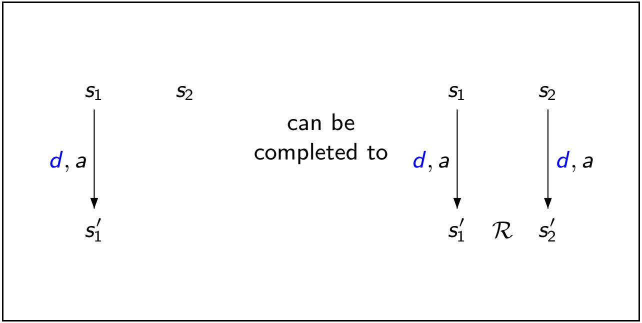
Simulation

- Example: t_0 simulates s_0



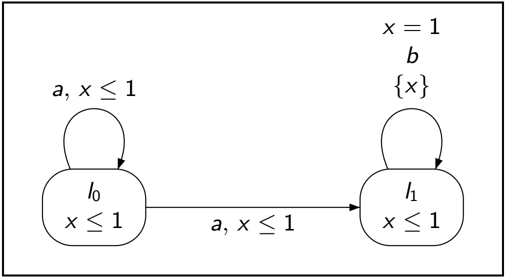
Timed simulation

- Timed transition system $(S, Act, \rightarrow)$, where $\rightarrow \subseteq S \times \mathbb{R}_{\geq 0} \times Act \times S$.
- Relation $\mathcal{R} \subseteq S \times S$ is a timed simulation relation if $\mathcal{R}$ satisfies the following condition:
  $(s_1, s_2) \in \mathcal{R}$ implies that, for each $s_1 \xrightarrow{d,a} s_1'$, there exists $s_2 \xrightarrow{d,a} s_2'$ such that $(s_1', s_2') \in \mathcal{R}$.



Timed automata

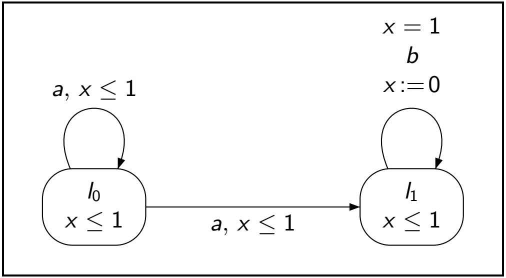
- Timed automata [AlurDill94]:
  - Finite-state graph (where the nodes are called *locations*).
  - Finite set of *clocks*: real-valued variables increasing at the same rate as real-time.
  - Clock constraints (*invariants* in locations, *guards* on edges).
  - Clock resets (set some clocks to 0 when an edge is traversed).



Timed automata

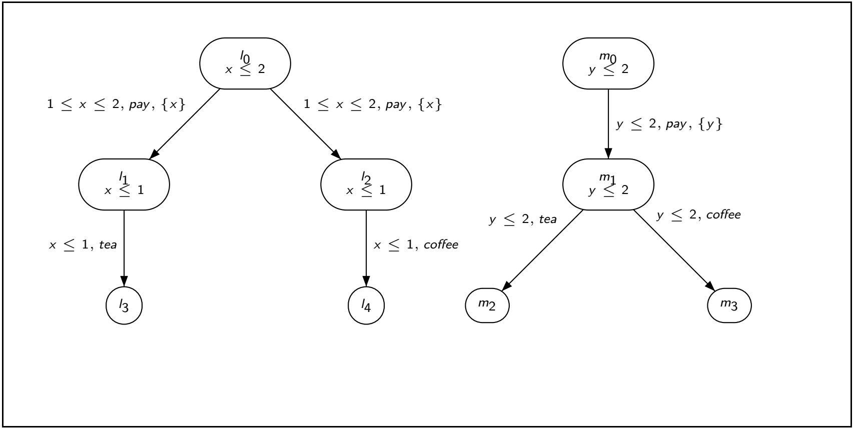
- Semantics of timed automata (in brief):
  - Represented by a timed transition system $(S, Act, \to)$, where $\to \subseteq S \times \mathbb{R}_{\geq 0} \times Act \times S$.
  - States: of the form (l, v), where l is a location and $v : \mathcal{X} \to \mathbb{R}_{\geq 0}$ is a *clock valuation* (must satisfy the invariant condition of l).
  - Transitions: for example (only a selection...),

$$((l_0, x = 0.2), 0.1, a, (l_0, x = 0.3)),\ ((l_0, x = 0.3), 0.7, a, (l_1, x = 1)),$$
$$((l_1, x = 1), 0, b, (l_1, x = 0)),\ ((l_1, x = 0), 1, b, (l_1, x = 0))$$



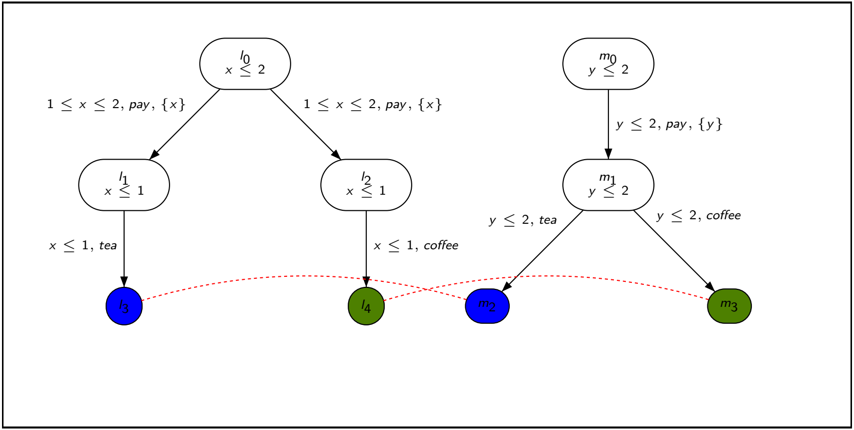
Timed simulation

- Example: does $(m_0, y = 0)$ timed simulate $(l_0, x = 0)$, given that $(m_2, ??)$ timed simulates $(l_3, ??)$ and $(m_3, ??)$ timed simulates $(l_4, ??)$?



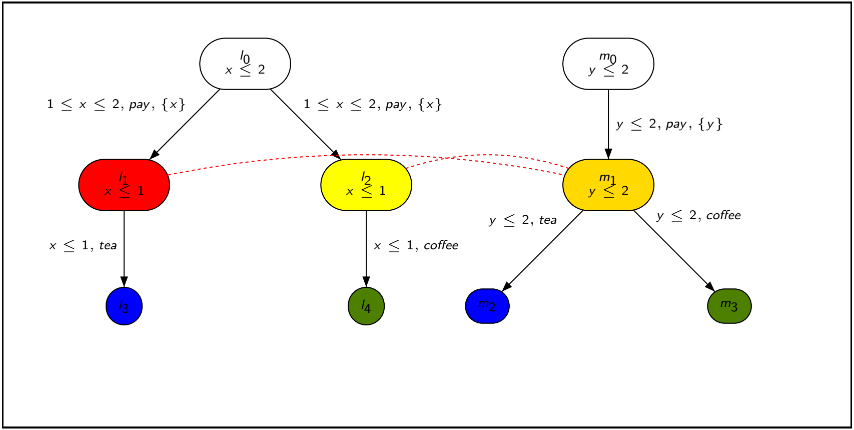
Timed simulation

- Example:



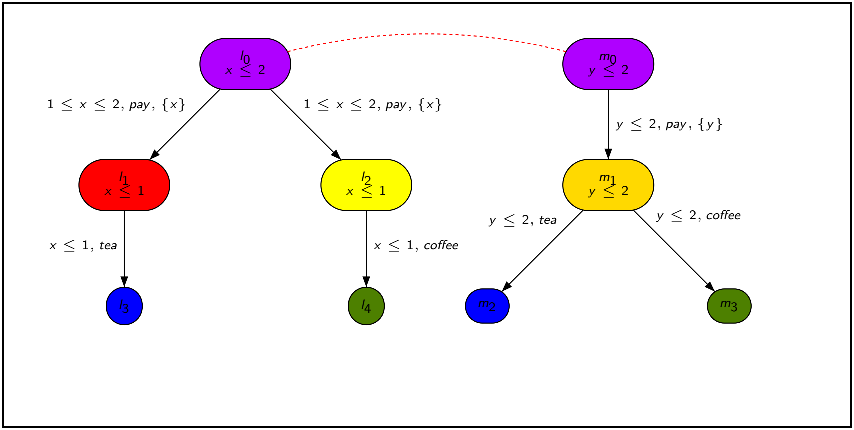
Timed simulation

- Example:
  $(m_1, y = 0)$ timed simulates $(l_1, x = 0)$ and $(l_2, x = 0)$.



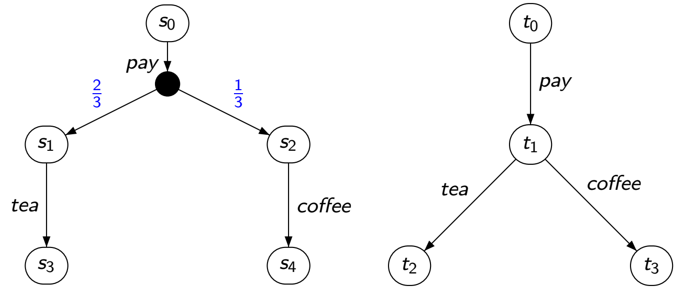
Timed simulation

- Example:
  $(m_0, y = 0)$ timed simulates $(l_0, x = 0)$.



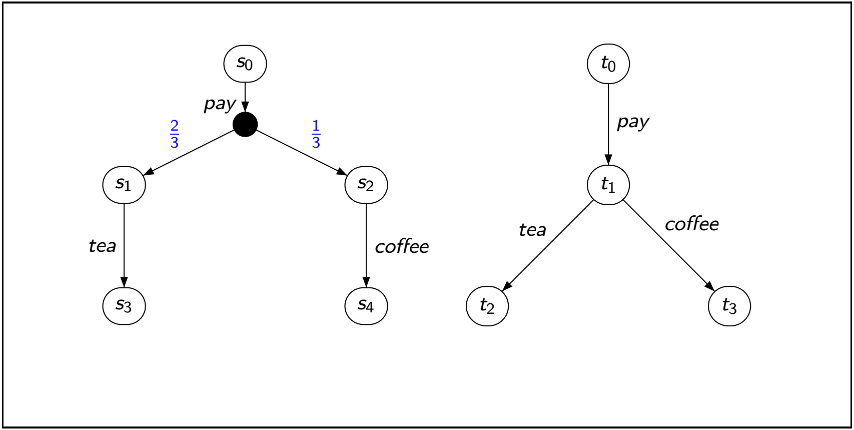
Simulation for probabilistic labelled transition systems

- Simulation for probabilistic labelled transition systems developed by [SegalaLynch95].



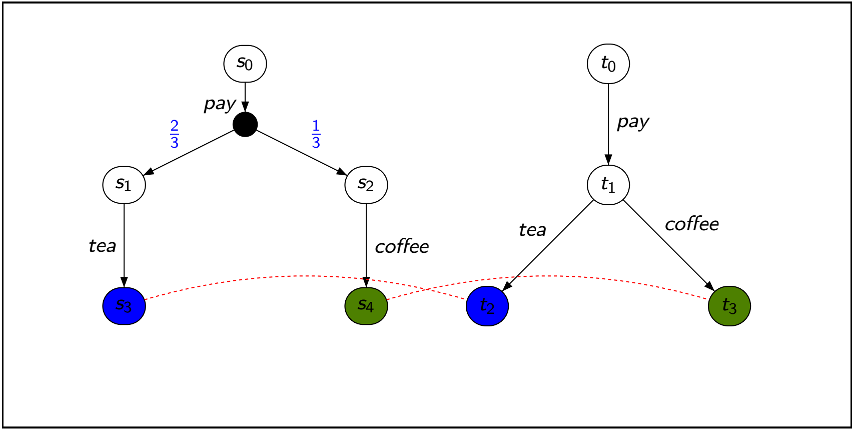
Simulation for probabilistic labelled transition systems

- Example: does t_0 simulate s_0, given that t_2 simulates s_3, and t_3 simulates s_4?



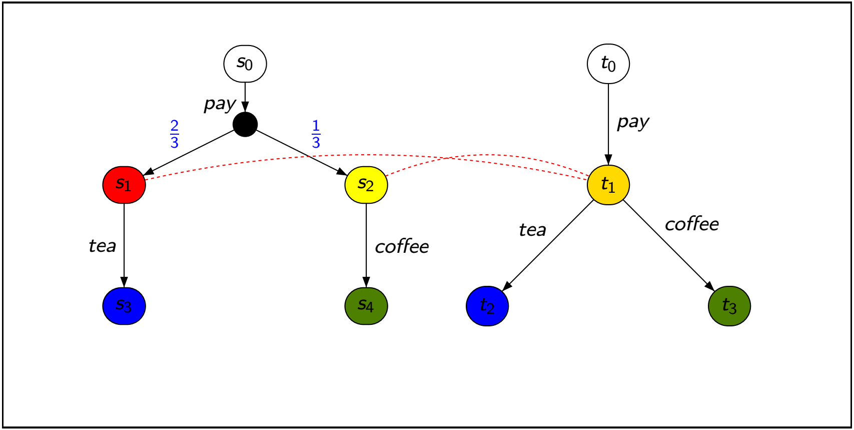
Simulation for probabilistic labelled transition systems

- Example:



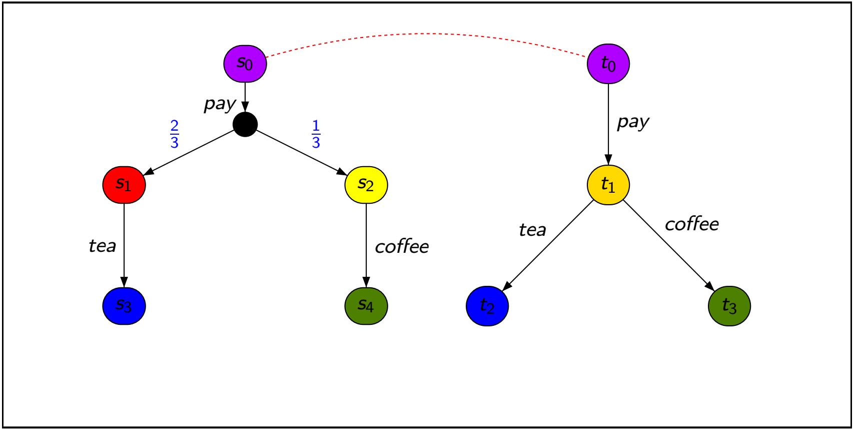
Simulation for probabilistic labelled transition systems

- Example:
  t_1 simulates s_1 and s_2.



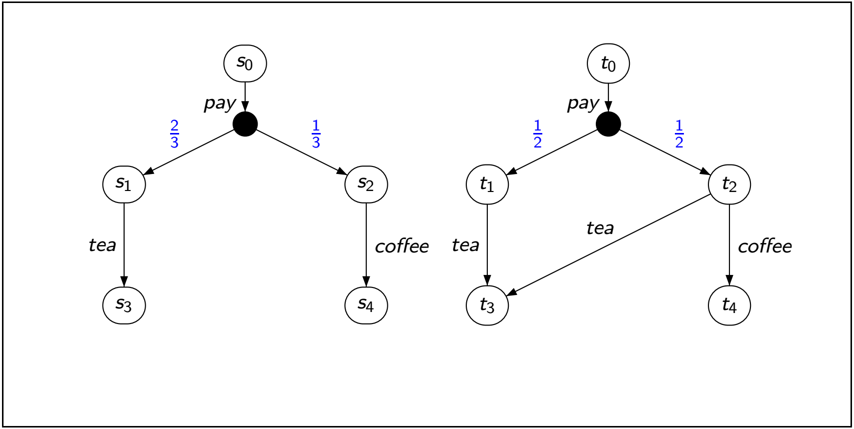
Simulation for probabilistic labelled transition systems

- Example:
  t_0 simulates s_0.



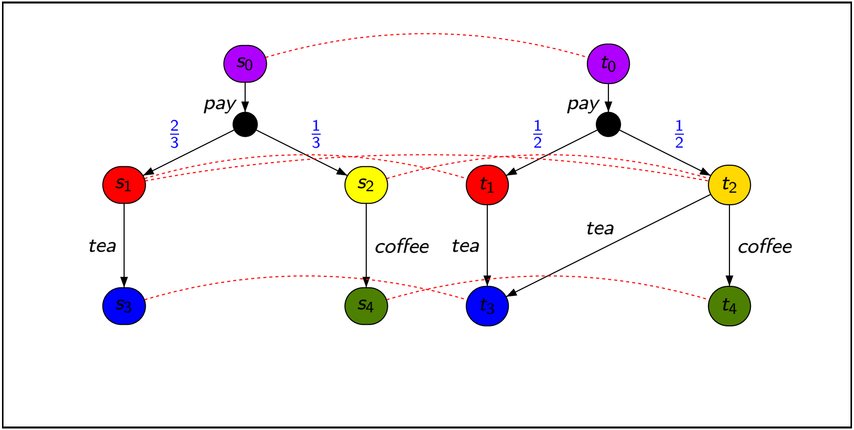
Simulation for probabilistic labelled transition systems

- Alternative example:
  does t_0 simulate s_0, given that t_2 simulates s_3, and t_3 simulates s_4?



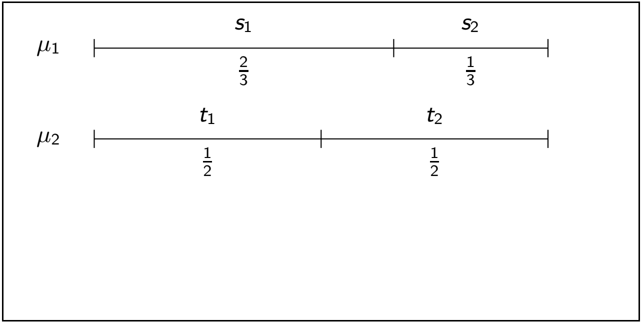
Simulation for probabilistic labelled transition systems

- Alternative example:
  t_0 simulates s_0.



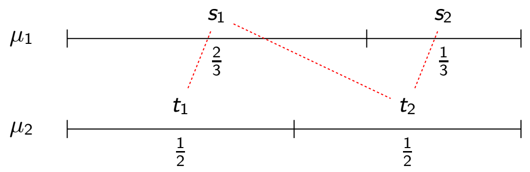
Simulation for probabilistic labelled transition systems

- $\mathrm{Dist}(S)$ is the set of probability distributions over S.
- Weight function [JonssonLarsen91]: for $\mu_1, \mu_2 \in \mathrm{Dist}(S)$ with respect to relation $\mathcal{R} \subseteq S \times S$ is a function $\Delta : S \times S \to [0, 1]$ such that:
  1. $\Delta(s_1, s_2) > 0$ implies $(s_1, s_2) \in \mathcal{R}$;
  2. $\sum_{s_2 \in S} \Delta(s_1, s_2) = \mu_1(s_1)$ for each $s_1 \in S$;
  3. $\sum_{s_1 \in S} \Delta(s_1, s_2) = \mu_2(s_2)$ for each $s_2 \in S$.
- Write $\mathit{weight}(\mu_1, \mu_2, \mathcal{R})$ if there is a weight function for μ_1, μ_2 w.r.t. $\mathcal{R}$
- Example:



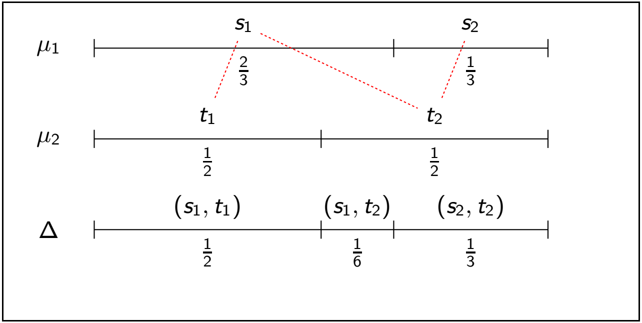
Simulation for probabilistic labelled transition systems

- $\mathsf{Dist}(S)$ is the set of probability distributions over S.
- Weight function [JonssonLarsen91]: for $\mu_1, \mu_2 \in \mathsf{Dist}(S)$ with respect to relation $\mathcal{R} \subseteq S \times S$ is a function $\Delta : S \times S \to [0, 1]$ such that:
  1. $\Delta(s_1, s_2) > 0$ implies $(s_1, s_2) \in \mathcal{R}$;
  2. $\sum_{s_2 \in S} \Delta(s_1, s_2) = \mu_1(s_1)$ for each $s_1 \in S$;
  3. $\sum_{s_1 \in S} \Delta(s_1, s_2) = \mu_2(s_2)$ for each $s_2 \in S$.
- Write *weight*$(\mu_1, \mu_2, \mathcal{R})$ if there is a weight function for μ_1, μ_2 w.r.t. $\mathcal{R}$
- Example:



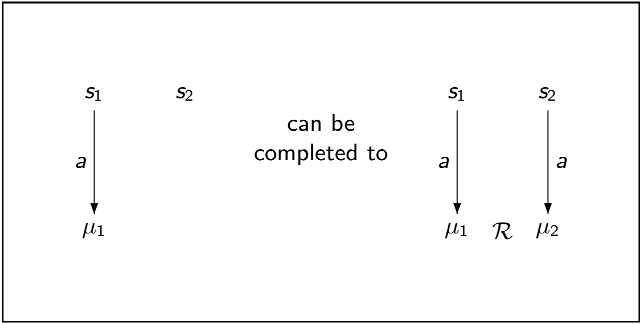
Simulation for probabilistic labelled transition systems

- $\mathrm{Dist}(S)$ is the set of probability distributions over S.
- Weight function [JonssonLarsen91]: for $\mu_1, \mu_2 \in \mathrm{Dist}(S)$ with respect to relation $\mathcal{R} \subseteq S \times S$ is a function $\Delta : S \times S \to [0, 1]$ such that:
  1. $\Delta(s_1, s_2) > 0$ implies $(s_1, s_2) \in \mathcal{R}$;
  2. $\sum_{s_2 \in S} \Delta(s_1, s_2) = \mu_1(s_1)$ for each $s_1 \in S$;
  3. $\sum_{s_1 \in S} \Delta(s_1, s_2) = \mu_2(s_2)$ for each $s_2 \in S$.
- Write $\mathit{weight}(\mu_1, \mu_2, \mathcal{R})$ if there is a weight function for μ_1, μ_2 w.r.t. $\mathcal{R}$
- Example:



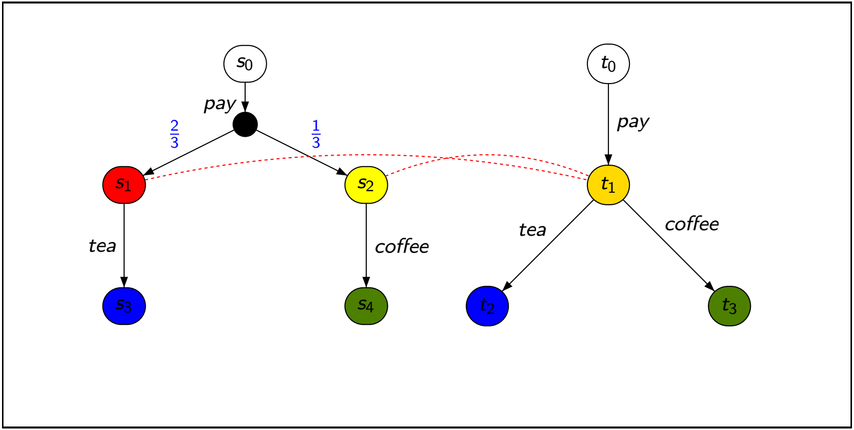
Simulation for probabilistic labelled transition systems

- Probabilistic labelled transition system (PLTS) $(S, Act, \rightarrow)$, where $\rightarrow \subseteq S \times Act \times$ Dist(S).
- Relation $\mathcal{R} \subseteq S \times S$ is a simulation relation [SegalaLynch95] if $\mathcal{R}$ satisfies the following condition:
  $(s_1, s_2) \in \mathcal{R}$ implies that, for each $s_1 \xrightarrow{a} \mu_1$, there exists $s_2 \xrightarrow{a} \mu_2$ such that *weight*$(\mu_1, \mu_2, \mathcal{R})$.



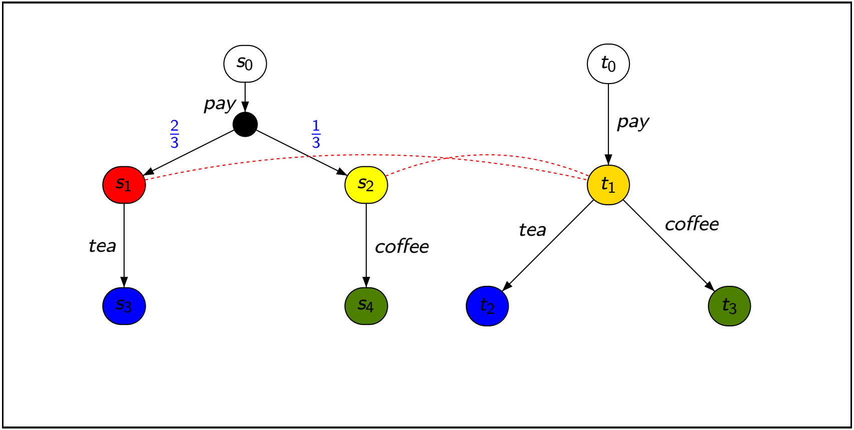
Simulation for probabilistic labelled transition systems

- Example:
  Does t_0 simulate s_0?



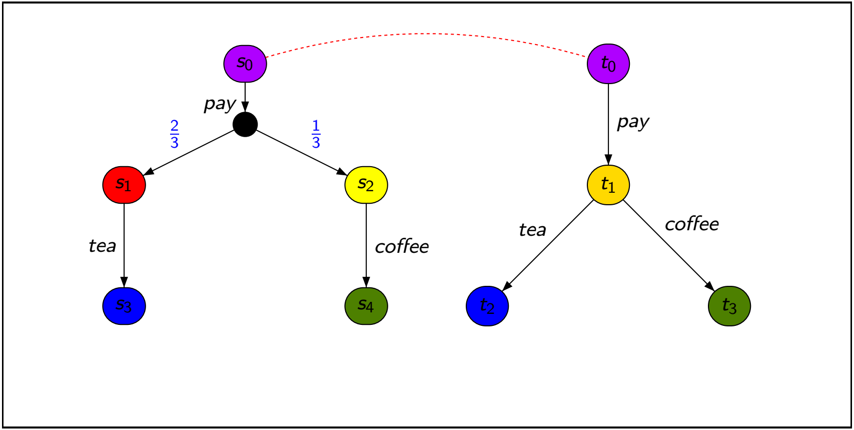
Simulation for probabilistic labelled transition systems

- Example: we have $(s_1, t_1), (s_2, t_1) \in \mathcal{R}$
  Weight function $\Delta(s_1, t_1) = \frac{2}{3}, \Delta(s_2, t_1) = \frac{1}{3}$



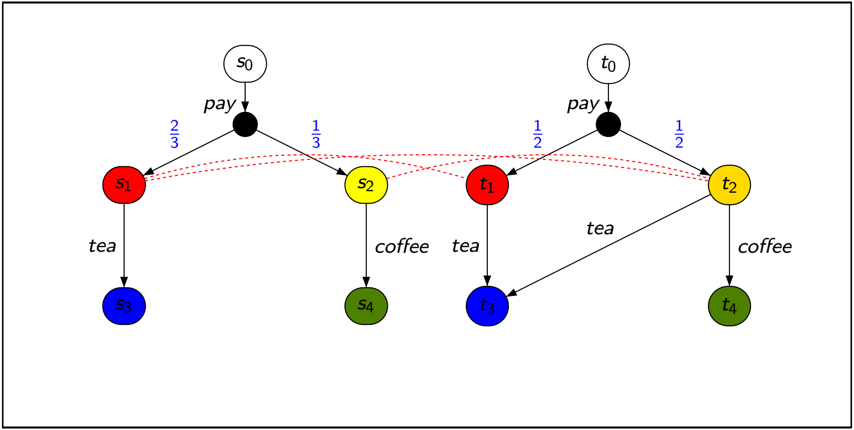
Simulation for probabilistic labelled transition systems

- Example:
  Hence t_0 simulates s_0



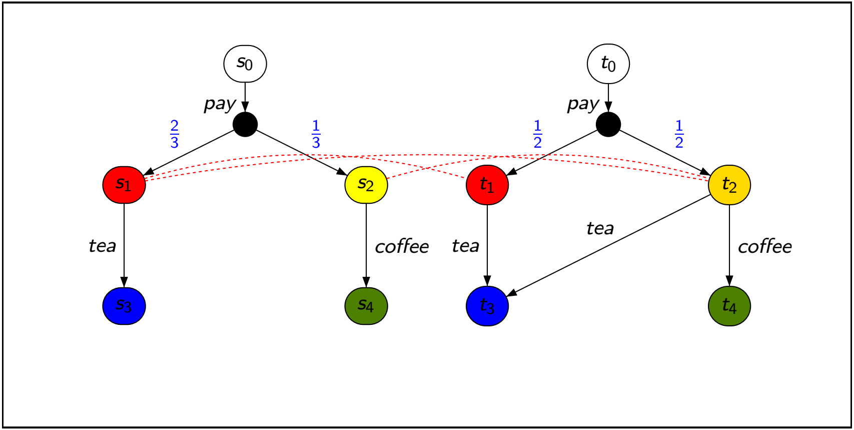
Simulation for probabilistic labelled transition systems

- Alternative example:
  Does t_0 simulate s_0?



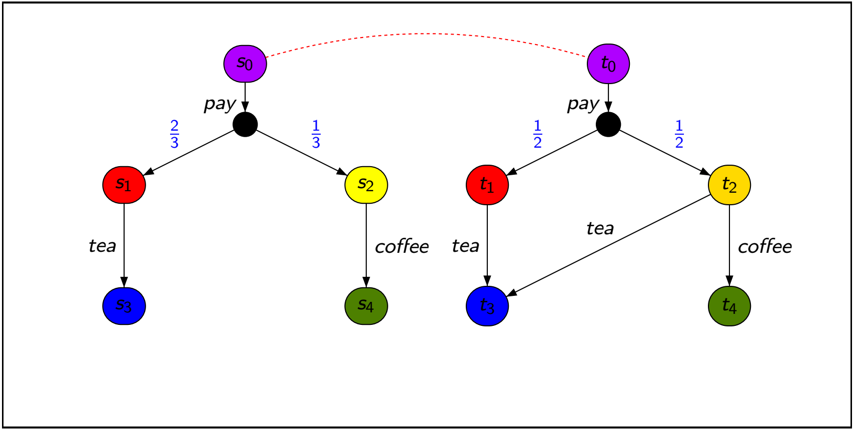
Simulation for probabilistic labelled transition systems

- Alternative example: we have $(s_1, t_1), (s_1, t_2), (s_2, t_2) \in \mathcal{R}$
  Weight function $\Delta(s_1, t_1) = \frac{1}{2}, \Delta(s_1, t_2) = \frac{1}{6}, \Delta(s_2, t_2) = \frac{1}{3}$



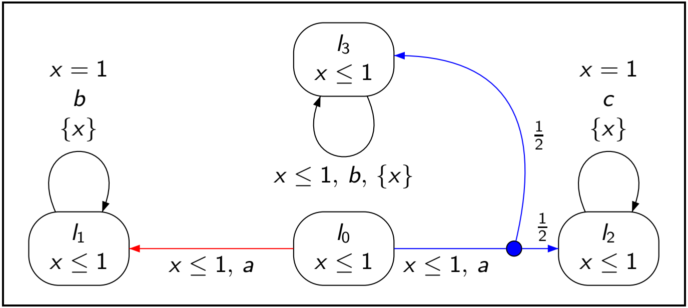
Simulation for probabilistic labelled transition systems

- Alternative example:
  Hence t_0 simulates s_0



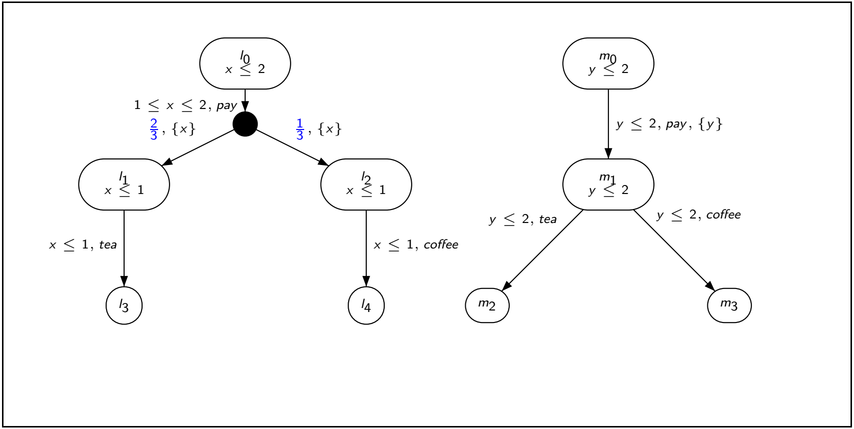
Probabilistic timed automata

- Probabilistic timed automata (PTA) [Jensen96,KNSS02]:
  - Timed automata plus probabilistic branching over "target edges" (target location, clock reset).
  - Semantics: in terms of timed probabilistic labelled transition systems.



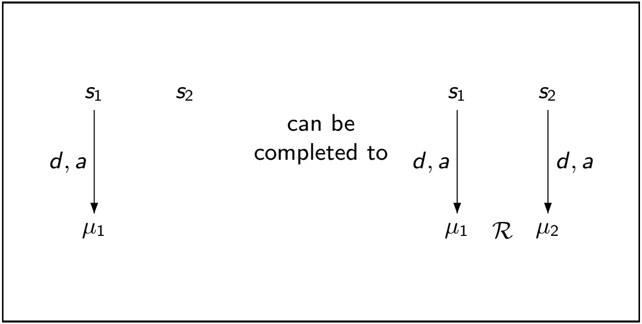
Simulation for probabilistic timed automata

- Example: does $(m_0, y = 0)$ timed simulate $(l_0, x = 0)$, given that $(m_2, ??)$ timed simulates $(l_3, ??)$ and $(m_3, ??)$ timed simulates $(l_4, ??)$?



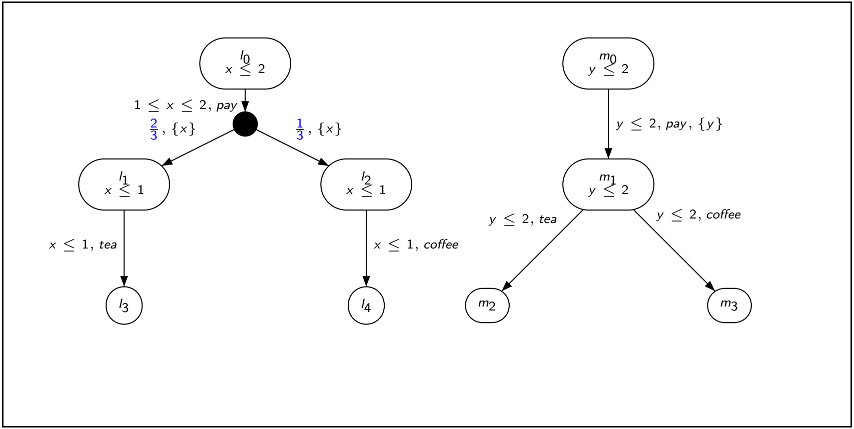
Timed simulation for timed PLTS

- Timed PLTS $(S, Act, \rightarrow)$, where $\rightarrow \subseteq S \times \mathbb{R}_{\geq 0} \times Act \times \mathrm{Dist}(S)$.
- Relation $\mathcal{R} \subseteq S \times S$ is a timed simulation relation if $\mathcal{R}$ satisfies the following condition:
  $(s_1, s_2) \in \mathcal{R}$ implies that, for each $s_1 \xrightarrow{d,a} \mu_1$, there exists $s_2 \xrightarrow{d,a} \mu_2$ such that $weight(\mu_1, \mu_2, \mathcal{R})$.



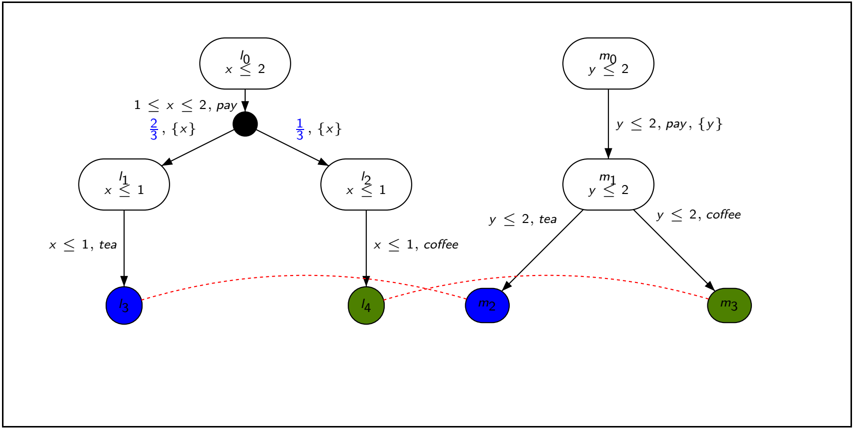
Simulation for probabilistic timed automata

- Example: does $(m_0, y = 0)$ timed simulate $(l_0, x = 0)$, given that $(m_2, ??)$ timed simulates $(l_3, ??)$ and $(m_3, ??)$ timed simulates $(l_4, ??)$?



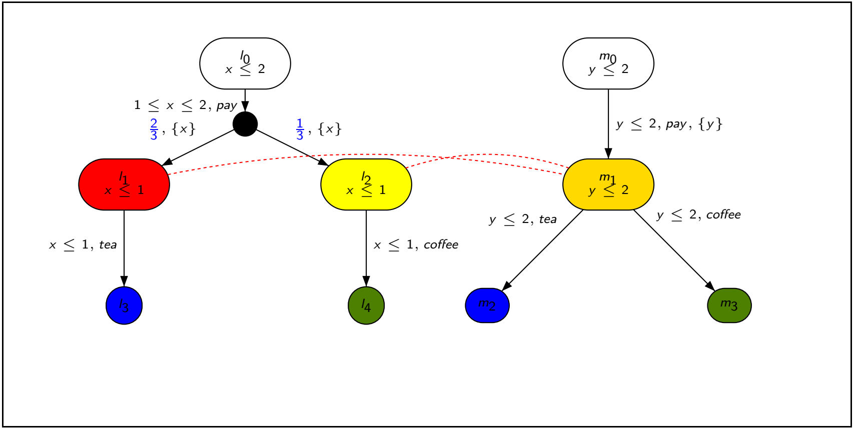
Timed simulation for PTA

- Example:



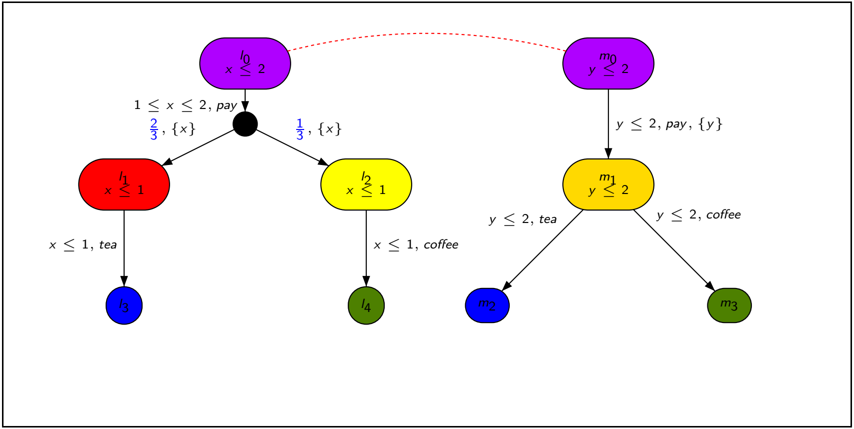
Timed simulation for PTA

- Example:
  $(m_1, y = 0)$ timed simulates $(l_1, x = 0)$ and $(l_2, x = 0)$
  I.e., $((l_1, x = 0), (m_1, y = 0)), ((l_2, x = 0), (m_1, y = 0)) \in \mathcal{R}$



Timed simulation for PTA

- Example: $(m_0, y = 0)$ timed simulates $(l_0, x = 0)$
  $\Delta((l_1, x = 0), (m_1, y = 0)) = \frac{2}{3}$
  $\Delta((l_2, x = 0), (m_1, y = 0)) = \frac{1}{3}$



Timed simulation for PTA: algorithm

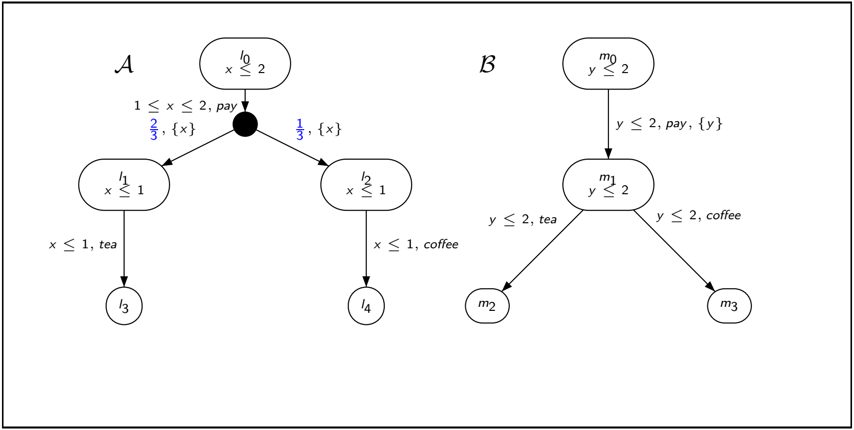
- Aim: to decide whether, given states $s_\mathcal{A}$ and $s_\mathcal{B}$ of two PTA $\mathcal{A}$, $\mathcal{B}$, respectively, whether $s_\mathcal{A}$ is timed simulated by $s_\mathcal{B}$.
- Combination of techniques for timed automata and for PLTS:
  - Timed automata: [TaşiranAKB96,BozzelliLP09] for timed simulation (based on [Čerāns92] for timed bisimulation).
  - PLTS: [BaierEM00, ZhangHEJ08].

Simulation for PLTS: algorithm

- Simulation algorithm for PLTS (for computing which states of PLTS $\mathcal{A}$ simulates which states of PLTS $\mathcal{B}$) [BaierEM00]:
  - Start by considering the relation $\mathcal{R} = S_{\mathcal{A}} \times S_{\mathcal{B}}$.
  - While possible, proceed by removing successively state pairs $(s_{\mathcal{A}}, s_{\mathcal{B}})$ from $\mathcal{R}$ if:
    $\exists s_{\mathcal{A}} \xrightarrow{a} \mu_{\mathcal{A}}$ such that $\nexists s_{\mathcal{B}} \xrightarrow{a} \mu_{\mathcal{B}}$ for which $\mathit{weight}(\mu_{\mathcal{A}}, \mu_{\mathcal{B}}, \mathcal{R})$.
  - If at some point no such state pair $(s_{\mathcal{A}}, s_{\mathcal{B}})$ exists, return the current $\mathcal{R}$.

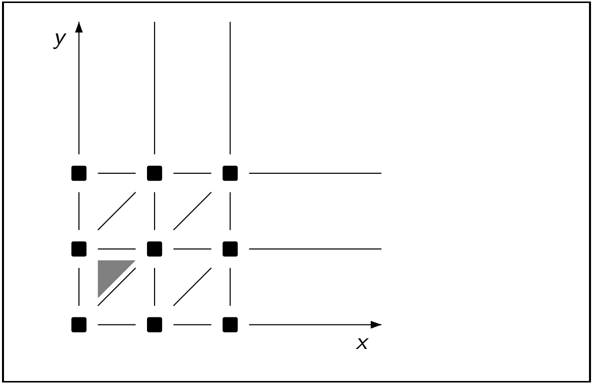
Timed simulation for PTA: algorithm

- Lift reasoning from states and transitions to regions and probability distributions over target edges (inspired by [Čerāns92,TașiranAKB96]).
- First construct region equivalence over *both* PTA $\mathcal{A}$ and $\mathcal{B}$.
- Example: PTA $\mathcal{A}$ has clock x, PTA $\mathcal{B}$ has clock y; maximal constant is 2.



Timed simulation for PTA: algorithm

- Construct regions over clock set $\{x, y\}$, with maximal constant 2.
- Example of region: $reg = ((l_1, m_2), 0 < x < y < 1)$.



Timed simulation for PTA: algorithm

Timed simulation is invariant over regions

If two states $(l_{\mathcal{A}}, v_{\mathcal{A}})$ and $(l_{\mathcal{B}}, v_{\mathcal{B}})$ in the same region *reg* are such that $(l_{\mathcal{B}}, v_{\mathcal{B}})$ timed simulates $(l_{\mathcal{A}}, v_{\mathcal{A}})$,
then *all* states $(l'_{\mathcal{A}}, v'_{\mathcal{A}})$ and $(l'_{\mathcal{B}}, v'_{\mathcal{B}})$ in *reg* are such that $(l'_{\mathcal{B}}, v'_{\mathcal{B}})$ timed simulates $(l'_{\mathcal{A}}, v'_{\mathcal{A}})$.

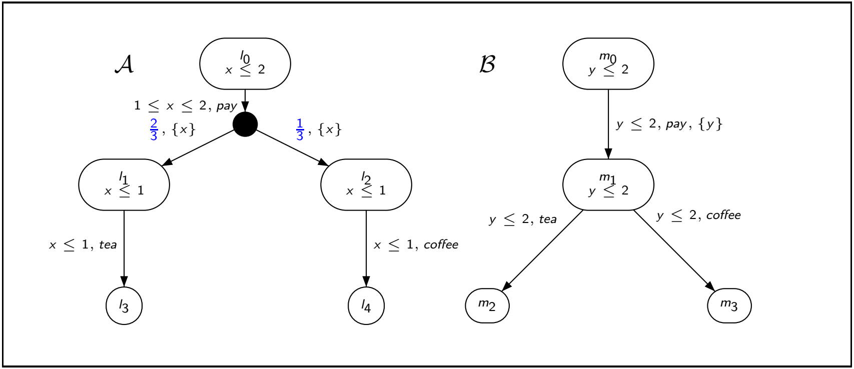
- Let $\mathcal{R} \subseteq S_{\mathcal{A}} \times S_{\mathcal{B}}$ be a relation which is invariant over regions.
- Then can represent $\mathcal{R}$ *symbolically* as a set Γ of regions: $reg \in \Gamma$ if and only if $((l_{\mathcal{A}}, v_{\mathcal{A}}), (l_{\mathcal{B}}, v_{\mathcal{B}})) \in \mathcal{R}$ for each $((l_{\mathcal{A}}, l_{\mathcal{B}}), v_{\mathcal{A}} \cdot v_{\mathcal{B}}) \in reg$.

Timed simulation for PTA: algorithm

- Simulation algorithm for PTA (for computing which states of PTA $\mathcal{A}$ simulates which states of PTA $\mathcal{B}$):
  - Start by considering Γ = Regions (which represents symbolically $\mathcal{R} = S_{\mathcal{A}} \times S_{\mathcal{B}}$).
  - While possible, proceed by removing successively regions *reg* from Γ if:
    $\exists (s_{\mathcal{A}}, s_{\mathcal{B}}) \in reg$ such that $\exists s_{\mathcal{A}} \xrightarrow{d,a} \mu_{\mathcal{A}}$ for which $\nexists s_{\mathcal{B}} \xrightarrow{d,a} \mu_{\mathcal{B}}$ such that $weight(\mu_{\mathcal{A}}, \mu_{\mathcal{B}}, \mathcal{R}_{\Gamma})$
    (where $\mathcal{R}_{\Gamma}$ is the relation represented symbolically by Γ).
  - If at some point no such region *reg* exists, return the current Γ.
- Problem: to check the condition, we need to check an infinite number of transitions, the distributions of which are over states.

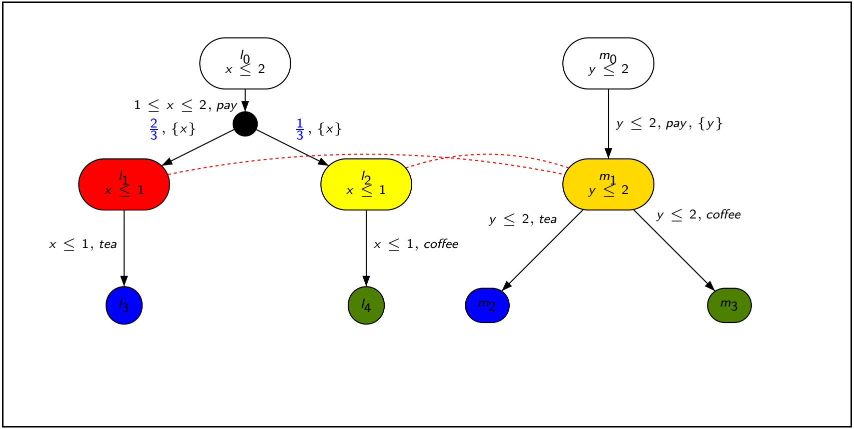
Timed simulation for PTA: algorithm

- Solution:
  - Consider only a finite number of time durations in transitions.
  - Lift Γ to the level of probability distributions over target edges in order to reason about probabilistic branching.
- Example: does $(m_0, y = 0.7)$ timed simulate $(l_0, x = 0.8)$?



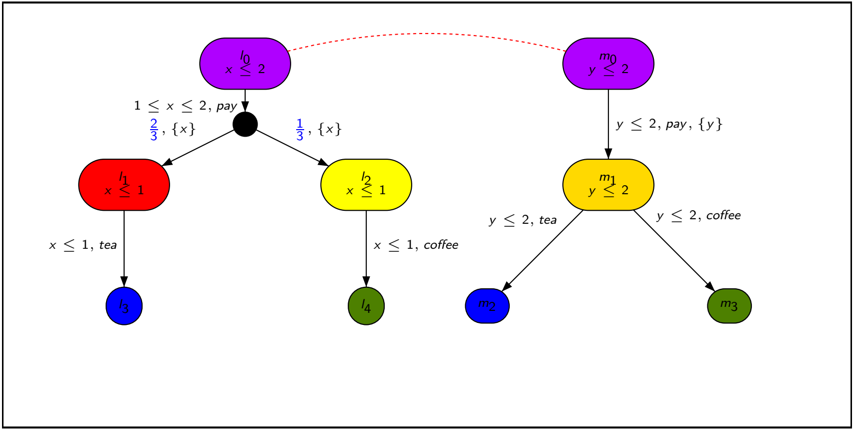
Timed simulation for PTA: algorithm

- Say we have computed Γ such that $((l_1, m_1), x = y = 0), ((l_1, m_2), x = y = 0) \in \Gamma$.
- Consider only durations $d \in \{0.2, 0.25, 0.3, 0.65, 1, 1.1, 1.2\}$.
- Weight function for distributions $p^{\mathcal{A}}$ and $p^{\mathcal{B}}$ of l_0 and m_0 w.r.t. a relation $\mathcal{E}$ on target edges which depends on Γ, $((l_0, m_0), 0 < y < x < 1)$ and d.



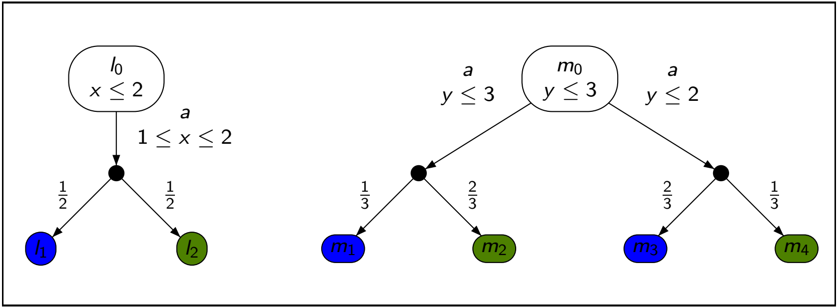
Timed simulation for PTA: algorithm

- Weight function for $p^{\mathcal{A}}$ and $p^{\mathcal{B}}$ with respect to $\mathcal{E}$: $\Delta(e^{\mathcal{A}}_{left}, e^{\mathcal{B}}) = \frac{2}{3}, \Delta(e^{\mathcal{A}}_{right}, e^{\mathcal{B}}) = \frac{1}{3}$.
- Hence $(m_0, y = 0.7)$ timed simulates $(l_0, x = 0.8)$.



Timed simulation for PTA: algorithm

- Checking whether a PTA $\mathcal{B}$ timed simulates a PTA $\mathcal{A}$ is EXPTIME-complete (lower bound from TA case [LaroussinieSchnoebelen00]).
- Extension to timed bisimulation: make symmetric the condition to check whether a region *reg* should be removed from the current set Γ of regions.
- Extension to probabilistic timed (bi)simulation [SegalaLynch95]: consider convex combinations of distributions in the condition to check whether a region *reg* should be removed from the current set Γ of regions.



Logical characterization

- Problem: identify a logic such that whenever two PTA states satisfy the same formulas of the logic, then the states are timed bisimilar.
- PTLogic: Henessey-Milner logic with:
  - Timed diamond modality [HolmerLY91,BozzelliLP09].
  - Probabilistic threshold operator [ParmaSegala07].
- Syntax of PTLogic:

$$\psi ::= \quad \mathit{true} \;\mid\; \neg\psi \;\mid\; \psi \wedge \psi \;\mid\; \langle a, \sim c\rangle\psi \;\mid\; [\psi]_p$$

where a is an action, $c \in \mathbb{R}_{\geq 0}$ is a constant, and $p \in [0,1]$ is a probability.

Logical characterization

- Semantics of PTLogic:

$\mu \models true$		
$\mu \models \neg\psi$	iff	$\mu \not\models \psi$
$\mu \models \psi_1 \wedge \psi_2$	iff	both $\mu \models \psi_1$ and $\mu \models \psi_2$
$\mu \models \langle a, \sim c\rangle\psi$	iff	for all $s \in \text{support}(\mu)$ there exists $(s, d, a, \mu') \in \longrightarrow$ such that $d \sim c$ and $\mu' \models \psi$
$\mu \models [\psi]_p$	iff	$\sum_{s \models \psi} \mu(s) \geq p$

where we write $s \models \psi$ if and only if $\{s \mapsto 1\} \models \psi$.

Logical characterization

Logical characterization of timed bisimulation for PTA

For each pair s, s' of states of a PTA, we have s and s' are timed bisimilar if and only if the set of PTLogic formulas satisfied in s equals the set of PTLogic formulas satisfied in s.

- PTLogic can be adapted to the case of probabilistic bisimulation, to give an analogous result.

Conclusions

- Deciding timed (bi)simulation between PTA is EXPTIME-complete.
- Known logical characterizations of timed bisimulation for timed automata and PLTS can be combined for provide a logical characterization of bisimulation for PTA.
- Future work:
  - Weak timed (bi)similarity (abstract from non-observable computation) for PTA.
  - Quantitative versions of (bi)simulation for PTA.
  - Implementation: from regions to zones.