Exact and Approximate Abstraction for Classes of Stochastic Hybrid Systems

Jeremy Sproston
University of Turin
AVOCS 2014
25th September 2014
Setting

• Focus on:
  – Stochastic hybrid systems:
    • Discrete and continuous components (hybrid).
    • Behaviour governed by probabilistic evolution.
  – Abstraction techniques: simplify the system to enable/aid analysis.

• System analysis:

```
System (SHS) → SHS satisfies \( \varphi \) → Probability
```

Property (\( \varphi \))
Hybrid automata

- Hybrid automaton (HA) [ACH+95]:
  - Finite control graph
    - finite set of real-valued variables
    - constraints on variables labelling the graph (guards, invariants, flows, resets).
- Behaviour of a hybrid automaton: *time elapse* and *edge* transitions.
Hybrid automata

\[ x \in [20, 25] \]
\[ switch\_off \]

\[ l_{on} \]
\[ \dot{x} \in [2, 3] \]
\[ x \in [10, 25] \]

\[ l_{off} \]
\[ \dot{x} \in [-2, -1] \]
\[ x \in [10, 50] \]

\[ x \in [10, 15] \]
\[ switch\_on \]

**Example**

\((l_{on}, x = 12)\)

Start in \(l_{on}\) with \(x = 12\).
**Example**

\[(l_{on}, x = 12) \xrightarrow{4} (l_{on}, x = 23)\]

[Time] 4 units of time pass, and \(x\) increases at the rate 2.75 (\(\in [2, 3]\)), to become 23.
Hybrid automata

Example

\[(l_{on}, x = 12) \xrightarrow{4} (l_{on}, x = 23) \xrightarrow{\text{switch-off}} (l_{off}, x = 23)\]

[Edge] Go from \(l_{on}\) to \(l_{off}\).
Hybrid automata

Example

\((l_{on}, x = 12) \xrightarrow{4} (l_{on}, x = 23) \xrightarrow{\text{switch-off}} (l_{off}, x = 23) \xrightarrow{9} (l_{off}, x = 13.1)\)

[Time] 9 units of time pass, and \(x\) decreases at rate \(-1.1\) (\(\in [-2, -1]\)), to become 13.1.
Hybrid automata

[Diagram with nodes and transitions labeled with conditions and actions]

**Example**

\[(l_{on}, x = 12) \xrightarrow{4} (l_{on}, x = 23) \xrightarrow{\text{switch}_\text{off}} (l_{off}, x = 23) \xrightarrow{9} (l_{off}, x = 13.1) \xrightarrow{\text{switch}_\text{on}} \ldots \]

[Edge] Go from \(l_{off}\) to \(l_{on}\), etc.
Consider systems with **nondeterministic** and **probabilistic** choice.

- E.g., in $s_0$, nondeterministic choice between red and blue transitions; if blue, then make a probabilistic choice.
• Strategy: resolution of nondeterministic choice.
  – Example 1: always red transitions.
  – Example 2: always blue transitions.
  – Example 3: red transitions until the 10th visit to $s_0$, then blue transitions.
• Consider maximum and minimum probabilities of satisfying a property over all strategies.

• $\text{MaxProb}(PS,s)(\varphi) = \sup_{\sigma \in \text{Strategies}(PS)} \text{Prob}(\sigma,s)(\varphi)$

• $\text{MinProb}(PS,s)(\varphi) = \inf_{\sigma \in \text{Strategies}(PS)} \text{Prob}(\sigma,s)(\varphi)$
• Abstraction for probabilistic systems: exact.
  – E.g., collapse equivalent states (typically use probabilistic bisimulation [LS91,SL95]).
  – $\text{MaxProb}(PS,s)(\varphi) = \text{MaxProb}(Abs(PS),s)(\varphi)$
  – $\text{MinProb}(PS,s)(\varphi) = \text{MinProb}(Abs(PS),s)(\varphi)$
  – Useful to reduce infinitary state spaces to finitary state spaces.
**Probabilistic systems**

- Abstraction for probabilistic systems: exact.
  - E.g., collapse equivalent states (typically use probabilistic bisimulation [LS91,SL95]).
  - MaxProb\((PS,s)\)(\(\varphi\))=MaxProb\((\text{Abs}(PS),s)\)(\(\varphi\))
  - MinProb\((PS,s)\)(\(\varphi\))=MinProb\((\text{Abs}(PS),s)\)(\(\varphi\))
  - Useful to reduce infinitary state spaces to finitary state spaces.
Abstraction for probabilistic systems: exact.
- E.g., collapse equivalent states (typically use probabilistic bisimulation [LS91,SL95]).
- $\text{MaxProb}(PS,s)(\varphi) = \text{MaxProb}(\text{Abs}(PS),s)(\varphi)$
- $\text{MinProb}(PS,s)(\varphi) = \text{MinProb}(\text{Abs}(PS),s)(\varphi)$
- Useful to reduce infinitary state spaces to finitary state spaces.
Abstraction for probabilistic systems: approximate.
- E.g., replace (some) probabilistic choice with nondeterministic choice.

\[
\begin{align*}
\text{MaxProb}(\text{PS}, s)(\varphi) & \leq \text{MaxProb}(\text{Abs}(\text{PS}), s)(\varphi) \\
\text{MinProb}(\text{PS}, s)(\varphi) & \geq \text{MinProb}(\text{Abs}(\text{PS}), s)(\varphi)
\end{align*}
\]
Probabilistic systems

- Abstraction for probabilistic systems: approximate.
  - E.g., replace (some) probabilistic choice with nondeterministic choice.

- \(\text{MaxProb}(\text{PS},s)(\varphi) \leq \text{MaxProb}(\text{Abs(PS)},s)(\varphi)\)
- \(\text{MinProb}(\text{PS},s)(\varphi) \geq \text{MinProb}(\text{Abs(PS)},s)(\varphi)\)
Probabilistic hybrid automata

- Probabilistic hybrid automaton (PHA):
  - Hybrid automaton + discrete probabilistic branching over edges.
  - Behaviour: time elapse and probabilistic edge transitions.
Probabilistic hybrid automata

Example

\((l_{on}, 12, 0)\)

Start in \(l_{on}\) with \(x = 12\) and \(y = 0\).
Probabilistic hybrid automata

Example

\[(l_{on}, 12, 0) \xrightarrow{\frac{4}{1}} (l_{on}, 23, 4)\]

[Time] 4 units of time pass, and \(x\) increases at the rate 2.75 \((\in [2, 3])\), and \(y\) increases at rate 1, with probability 1.
Probabilistic hybrid automata

Example

\[(l_{on}, 12, 0) \xrightarrow{\frac{4}{1}} (l_{on}, 23, 4) \xrightarrow{\text{switch\_off}} \frac{1}{1} (l_{off}, 23, 4)\]

[Probabilistic edge] Go from \(l_{on}\) to \(l_{off}\) with probability 1.
Probabilistic hybrid automata

Example

\[
(l_{on}, 12, 0) \xrightarrow{4}{1} (l_{on}, 23, 0) \xrightarrow{1}{1} (l_{off}, 23, 4) \xrightarrow{9}{1} (l_{off}, 13.1, 13)
\]

[Time] 9 units of time pass, and \( x \) decreases at rate -1.1 \((\in [-2, -1])\), and \( y \) increases at rate 1, with probability 1.
Probabilistic hybrid automata

Example

Probabilistic edge] Go from $l_{off}$ to $l_{on}$ with probability $\frac{99}{100}$, and from $l_{off}$ to $l_{malfunction}$ (setting $y$ to 0) with probability $\frac{1}{100}$.
Exact abstraction

• Overall aim: unify and generalise isolated results on exact abstraction for PHA.

• Previous work: exact abstraction for subclasses of PHA using probabilistic bisimulation [Spr00,KNSS02,Spr11].
  – Reduce infinite-state probabilistic system semantics of a PHA to a finite-state probabilistic system.
  – Subclasses of (P)HA: characterised by the form of conditions on continuous variables (guards, invariants, flows, resets).

• Form of previous results: probabilistic generalisations of showing that all instances of a subclass of HA have a finite number bisimulation quotient.
Bisimulation

• Bisimulation: if equivalence relation $R$ on states is such that, for each $(s,s') \in R$, we have that:
  – for all transitions $s \xrightarrow{a} t$, there exists transition $s' \xrightarrow{a} t'$ such that $(t,t') \in R$
then $R$ is a bisimulation.
Probabilistic bisimulation

- Probabilistic bisimulation: if equivalence relation $R$ on states is such that, for each $(s, s') \in R$, we have that:
  - for all transitions $s \xrightarrow{a} \mu$, there exists transition $s' \xrightarrow{a} \mu'$ such that $\mu$ and $\mu'$ assign the same total probability to each equivalence class of $R$ then $R$ is a probabilistic bisimulation.
Exact abstraction

• Approach: given PHA $P$, construct a HA $\text{nonprob}(P)$ with the same state space, such that bisimilar states of $\text{nonprob}(P)$ are also probabilistic bisimilar in $P$.
• Encode within actions information on distributions, resets and target locations.
  – Given $\text{nonprob}(P)$, can reconstruct $P$. 
Exact abstraction

PHA $P$

guard $g$

action $a$

prob $\lambda$, reset $r$

location $loc$

flow dynamics

guard $g'$

action $a'$

prob $\lambda'$, reset $r'$

Distribution $p$
(with $p(r, loc) = \lambda$)
Exact abstraction

HA \textit{nonprob}(P)

Guard $g$
Action $\langle a, p, r, \text{loc} \rangle$
Reset $r$

Location $\text{loc}$
Flow dynamics

...
Exact abstraction

- Consider bisimilar states \((\text{loc, val}), (\text{loc'}, \text{val'})\) of \(\text{nonprob}(P)\).
- Probabilistic bisimilarity of \((\text{loc, val}), (\text{loc'}, \text{val'})\) in \(P\):
Exact abstraction

- Consider bisimilar states \((\text{loc}, \text{val}), (\text{loc}', \text{val}')\) of \text{nonprob}(P).
- Probabilistic bisimilarity of \((\text{loc}, \text{val}), (\text{loc}', \text{val}')\) in \(P:\)
Exact abstraction

• Consider bisimilar states $(\text{loc}, \text{val})$, $(\text{loc}', \text{val}')$ of $\text{nonprob}(P)$.
• Probabilistic bisimilarity of $(\text{loc}, \text{val})$, $(\text{loc}', \text{val}')$ in $P$: 

![Diagram](image-url)
Exact abstraction

- Consider bisimilar states \((\text{loc}, \text{val}), (\text{loc}', \text{val}')\) of \(\text{nonprob}(P)\).
- Probabilistic bisimilarity of \((\text{loc}, \text{val}), (\text{loc}', \text{val}')\) in \(P\):

\[
\begin{align*}
nonprob(P) & \quad P \\
(\text{loc}, \text{val}) & \quad (\text{loc}, \text{val}) \\
(\text{loc}', \text{val}') & \quad (\text{loc}', \text{val}') \\
\lambda_1 & \quad \lambda_1 \\
\end{align*}
\]
Exact abstraction

- Consider bisimilar states \((\text{loc}, \text{val}), (\text{loc'}, \text{val'})\) of \(\text{nonprob}(P)\).
- Probabilistic bisimilarity of \((\text{loc}, \text{val}), (\text{loc'}, \text{val'})\) in \(P\):

\[
p(r_2, \text{loc}_2) = \lambda_2
\]
Exact abstraction

- Consider bisimilar states \((\text{loc}, \text{val}), (\text{loc}', \text{val}')\) of \(\text{nonprob}(P)\).
- Probabilistic bisimilarity of \((\text{loc}, \text{val}), (\text{loc}', \text{val}')\) in \(P\):

```
\text{nonprob}(P)
```

```
\text{P}
```

\[
p(r_3, \text{loc}_3) = \lambda_3
\]
Exact abstraction

• Consider bisimilar states \((\text{loc}, \text{val}), (\text{loc}', \text{val}')\) of \(\text{nonprob}(P)\).
• Probabilistic bisimilarity of \((\text{loc}, \text{val}), (\text{loc}', \text{val}')\) in \(P\):

\[
\nonumber p(r_1, \text{loc}_1) = \lambda_1
\]

\[
\nonumber p(r_1, \text{loc}_1) = \lambda_1
\]
Exact abstraction

- Consider bisimilar states \((loc, val), (loc', val')\) of \(\text{nonprob}(P)\).
- Probabilistic bisimilarity of \((loc, val), (loc', val')\) in \(P\):

\[
p(r_2, loc_2) = \lambda_2
\]
Exact abstraction

- Consider bisimilar states \((\text{loc}, \text{val}), (\text{loc}', \text{val}')\) of \(\text{nonprob}(P)\).
- Probabilistic bisimilarity of \((\text{loc}, \text{val}), (\text{loc}', \text{val}')\) in \(P\):

\[
\begin{align*}
\text{nonprob}(P) & \\
(\text{loc}, \text{val}) & \quad (\text{loc}', \text{val}') \\
\text{a} & \quad \text{a} \\
\lambda_1 & \quad \lambda_1 \\
\lambda_2 & \quad \lambda_2 \\
\lambda_3 & \quad \lambda_3 \\
\end{align*}
\]
Exact abstraction

- Consider bisimilar states \((\text{loc}, \text{val}), (\text{loc’}, \text{val’})\) of \(\text{nonprob}(P)\).
- Probabilistic bisimilarity of \((\text{loc}, \text{val}), (\text{loc’}, \text{val’})\) in \(P\):
Exact abstraction

• If \( \text{nonprob}(P) \) has a finite bisimulation quotient, then \( P \) has a finite number of probabilistic bisimulation quotient.

• For any subclass of HA containing only HA with finite bisimulation quotients, their associated class of PHA contains only PHA with a finite probabilistic bisimulation quotients.
Approximate abstraction

- Extend PHA with probabilistic choice over flows (variable evolution as time passes)?
  - E.g., thermostat: probabilistic choice of flow of $x$ (temperature), following a truncated normal distribution within $[1,6]$ or $[-4,-1]$. 

$$
\begin{align*}
\text{l}_{\text{deact}} & : \begin{cases}
\dot{x} = 0, \dot{y} = 1 \\
10 \leq x \leq 30
\end{cases} \\
\text{l}_{\text{on}} & : \begin{cases}
\dot{x} \in [1,6], \dot{y} = 1 \\
10 \leq x \leq 25
\end{cases} \\
\text{l}_{\text{off}} & : \begin{cases}
\dot{x} \in [-4,-1], \dot{y} = 1 \\
10 \leq x \leq 30
\end{cases} \\
\text{l}_{\text{malf}} & : \begin{cases}
\dot{x} \in [1,6], \dot{y} = 1 \\
10 \leq x \leq 30, y \leq 20
\end{cases}
\end{align*}
$$
Approximate abstraction

• Consider: the choice of the flow is made on entry to the location.
• Can represent this case using *stochastic hybrid automata (SHA)* [FHH+11, Hahn13].
  – Variables can be reset according to continuous probability distributions.

\[ \text{Unif}(1,3) \rightarrow \dot{z} = c, \dot{c} = 0 \]
Approximate abstraction

- Approximate analysis of SHA: construction of an approximate PHA [FHH+11, Hahn13] (via [KNSS00]).
  - Replace probabilistic choice with nondeterministic choice.

\[
\begin{align*}
\frac{1}{2} &
\text{c}\':\in[1,2] \\
\frac{1}{2} &
\text{c}\':\in[2,3] \\
\dot{z} &= c, \dot{c} = 0
\end{align*}
\]
Approximate abstraction

• Thermostat example is a probabilistic rectangular automaton (PRA):
  – Subclass of PHA with constraints of the form $x \in [a,b]$ (guards, invariants), $x: \in [a,b]$ (resets) and $\dot{x} \in [a,b]$ (flows) (for integers $a$, $b$)
  – Restricting PRA to discrete-time results in a finite number of probabilistic bisimulation classes [Spr11].
    • Can construct an exact (w.r.t. discrete-time PRA) finite-state probabilistic system directly, avoiding the further approximation step of [ZSR+12].
  – But flows such as $\dot{z}=c$, $\dot{c}=0$ are not allowed in PRA (c is a variable).
Approximate abstraction

\[ \dot{z} = c, \dot{c} = 0 \]

\[ \frac{1}{2} c' : \in [1,2] \]

\[ \dot{z} \in [1,2] \]

\[ \frac{1}{2} c' : \in [2,3] \]

\[ \dot{z} \in [2,3] \]
Approximate abstraction

\[ \frac{\gamma}{2}, \quad c' \in [1, 2] \]

\[ \frac{\gamma}{2}, \quad c' \in [2, 3] \]

\[ \dot{z} = c, \quad \dot{c} = 0 \]

Good: the bottom PHA is a PRA.
Bad: it is not probabilistic bisimilar to the top PHA.

\[ \frac{\gamma}{2} \]

\[ \dot{z} \in [1, 2] \]

\[ \frac{\gamma}{2} \]

\[ \dot{z} \in [2, 3] \]
Approximate abstraction

\[ \frac{1}{2} \]
\[ c' : \in [1,2] \]

\[ \dot{z} = c, \quad \dot{c} = 0 \]

\[ \frac{1}{2} \]
\[ c' : \in [2,3] \]

\[ \dot{z} \in [1,2] \]

\[ \dot{z} \in [2,3] \]

Good: the bottom PHA is a PRA.
Bad: it is not probabilistic bisimilar to the top PHA.
Good: max/min reachability probabilities are the same in the top and bottom PHA.
Approximate abstraction

- This framework can include periodic resampling of flows every $k$ time units, e.g.:

\[ \dot{z} = c, \ \dot{c} = 0 \]

\[ x = k \]

\[ c' : \in \text{Unif}(1,3) \]

\[ x' = 0 \]
Approximate abstraction

- Thermostat example: with periodic resampling every $1/10$ time units.
  - Max/min probability to reach deactivation location within $T$ time units.
  - “Original” features nondeterministic choice over flows.
  - “New” considers use of the truncated normal distributions (approx. with 3 intervals).
  - Finite-state probabilistic systems coded directly in PRISM [KNP11].

![Graph showing the probability of deactivation over time](image)
Conclusions

• Exact: for any class of HA with finite bisimulation quotient, the associated PHA class will have a finite probabilistic bisimulation quotient.

• Approximate: an attempt to incorporate some probabilistic choice over flows in PRA, combined with existing approximation of [FHH+11,Hahn13].
  – More realistic case studies?