Model Checking for Probabilistic Hybrid Systems

Marta Kwiatkowska, Ernst Moritz Hahn
Oxford University Computing Laboratory

Holger Hermanns, Arnd Hartmanns
Saarland University, Dependable Systems and Software

CPSWeek’13, Philadelphia, April 2013
Part 2

Probabilistic Hybrid Systems
• **Markov decision processes (MDPs)**
  – both probability and nondeterminism
  – in a state, there is a nondeterministic choice between multiple probability distributions over successor states

• **Adversaries**
  – resolve nondeterministic choices based on history so far
  – properties quantify over all possible adversaries
  – e.g. $P_{<0.1}[\Diamond_{\text{err}}]$ – maximum probability of an error is $< 0.1$
Real-world protocol examples

• Systems with probability, nondeterminism and real-time
  – e.g. communication protocols, randomised security protocols

• Randomised back-off schemes
  – Ethernet, WiFi (802.11), Zigbee (802.15.4)

• Random choice of waiting time
  – Bluetooth device discovery phase
  – Root contention in IEEE 1394 FireWire

• Random choice over a set of possible addresses
  – IPv4 dynamic configuration (link-local addressing)

• Random choice of a destination
  – Crowds anonymity, gossip-based routing
Overview (Part 2)

- **Time, clocks and zones**
- Probabilistic timed automata (PTAs)
  - definition, examples, semantics, reachability
- Model checking for PTAs
  - digital clocks
  - zone-based approaches
  - forwards reachability
- Probabilistic hybrid automata (PHAs)
  - definition, examples, semantics, extensions
• Dense time domain: non-negative reals $\mathbb{R}_{\geq 0}$
  – from this point on, we will abbreviate $\mathbb{R}_{\geq 0}$ to $\mathbb{R}$

• Finite set of clocks $x \in X$
  – variables taking values from time domain $\mathbb{R}$
  – increase at the same rate as real time

• A clock valuation is a tuple $v \in \mathbb{R}^X$. Some notation:
  – $v(x)$: value of clock $x$ in $v$
  – $v+t$: time increment of $t$ for $v$
    • $(v+t)(x) = v(x)+t \ \forall x \in X$
  – $v[Y:=0]$: clock reset of clocks $Y \subseteq X$ in $v$
    • $v[Y:=0](x) = 0$ if $x \in Y$ and $v(x)$ otherwise
Zones (clock constraints)

- Zones (clock constraints) over clocks $X$, denoted $\text{Zones}(X)$:

  $$\zeta ::= x \leq d \mid c \leq x \mid x+c \leq y+d \mid \neg\zeta \mid \zeta \lor \zeta$$

  - where $x, y \in X$ and $c, d \in \mathbb{N}$
  - used for both syntax of PTAs/properties and algorithms

- Can derive:
  - logical connectives, e.g. $\zeta_1 \land \zeta_2 \equiv \neg(\neg\zeta_1 \lor \neg\zeta_2)$
  - strict inequalities, through negation, e.g. $x > 5 \equiv \neg(x \leq 5)$...

- Some useful classes of zones:
  - closed: no strict inequalities (e.g. $x > 5$)
  - diagonal-free: no comparisons between clocks (e.g. $x \leq y$)
  - convex: define a convex set, efficient to manipulate
A clock valuation \( v \) satisfies a zone \( \zeta \), written \( v \triangleright \zeta \) if
- \( \zeta \) resolves to true after substituting each clock \( x \) with \( v(x) \)

The semantics of a zone \( \zeta \in \text{Zones}(X) \) is the set of clock valuations which satisfy it (i.e. a subset of \( \mathbb{R}^X \))
- NB: multiple zones may have the same semantics
- e.g. \( (x \leq 2) \land (y \leq 1) \land (x \leq y + 2) \) and \( (x \leq 2) \land (y \leq 1) \land (x \leq y + 3) \)

We consider only canonical zones
- i.e. zones for which the constraints are as ‘tight’ as possible
- \( O(|X|^3) \) algorithm to compute (unique) canonical zone [Dil89]
- allows us to use syntax for zones interchangeably with semantic, set-theoretic operations
• Clock valuations \( v \) and \( v' \) are c-equivalent if for any \( x,y \in X \)
  – either \( v(x) = v'(x) \), or \( v(x) > c \) and \( v'(x) > c \)
  – either \( v(x) - v(y) = v'(x) - v'(y) \) or \( v(x) - v(y) > c \) and \( v'(x) - v'(y) > c \)

• The c-closure of the zone \( \zeta \), denoted \( \text{close}(\zeta,c) \), equals
  – the greatest zone \( \zeta' \supseteq \zeta \) such that, for any \( v' \in \zeta' \),
    there exists \( v \in \zeta \) and \( v \) and \( v' \) are c-equivalent
  – c-closure ignores all constraints which are greater than \( c \)
  – for a given \( c \), there are only a finite number of c-closed zones
Operations on zones – Set theoretic

- Intersection of two zones: $\zeta_1 \cap \zeta_2$
- Similar for other operators
  - Union and difference of two zones: $\zeta_1 \cup \zeta_2$, $\zeta_1 \setminus \zeta_2$
  - Valuations obtained from by resetting the clocks in $Y$: $\zeta[Y:=0]$
  - Valuations which are in $\zeta$ if the clocks in $Y$ are reset: $[Y:=0]\zeta$
  - Forwards diagonal projection: $\rightarrow \zeta$
Overview (Part 2)

- Time, clocks and zones
- **Probabilistic timed automata (PTAs)**
  - definition, examples, semantics, reachability
- Model checking for PTAs
  - zone-based approaches
  - forwards reachability
- **Probabilistic hybrid automata (PHAs)**
  - definition, examples, semantics, extensions
Probabilistic timed automata (PTAs)

- Probabilistic timed automata (PTAs)
  - Markov decision processes (MDPs) + real-valued clocks
  - or: timed automata + discrete probabilistic choice
  - model probabilistic, nondeterministic and timed behaviour

- Syntax: A PTA is a tuple \((\text{Loc}, \text{l}_{\text{init}}, \text{Act}, \text{X}, \text{inv}, \text{prob}, \text{L})\)
  - \text{Loc} is a finite set of locations
  - \(\text{l}_{\text{init}} \in \text{Loc}\) is the initial location
  - \text{Act} is a finite set of actions
  - \text{X} is a finite set of clocks
  - \text{inv} : \text{Loc} \rightarrow \text{Zones}(\text{X})
    is the invariant condition
  - \text{prob} \subseteq \text{Loc} \times \text{Zones}(\text{X}) \times \text{Dist}(\text{Loc} \times 2^\text{X})
    is the probabilistic edge relation
  - \text{L} : \text{Loc} \rightarrow \text{AP} is a labelling function
Probabilistic edge relation

- **Probabilistic edge relation**
  - \( \text{prob} \subseteq \text{Loc} \times \text{Zones}(X) \times \text{Act} \times \text{Dist}(\text{Loc} \times 2^X) \)

- **Probabilistic edge \((l,g,a,p) \in \text{prob}\)**
  - \( l \) is the source location
  - \( g \) is the guard
  - \( a \) is the action
  - \( p \) target distribution

- **Edge \((l,g,a,p,l',Y)\)**
  - from probabilistic edge \((l,g,a,p)\) where \( p(l',Y) > 0 \)
  - \( l' \) is the target location
  - \( Y \) is the set of clocks to be reset
• Models a simple probabilistic communication protocol
  – starts in location $di$; after between 1 and 2 time units, the protocol attempts to send the data:
    • with probability 0.9 data is sent correctly, move to location $sr$
    • with probability 0.1 data is lost, move to location $si$
  – in location $si$, after 2 to 3 time units, attempts to resend
    • correctly sent with probability 0.95 and lost with probability 0.05
PTA Modelling

• Simple extension of guarded commands:
  – new variable type clock
  – new language construct invariant

• Invariants:
  – specified restrictions in clocks of a given module depending on its discrete variables
  – for parallel composition: conjunction of invariants is used

    module ptaexample
    s : [0..2] init 0;
x : clock;
invariant
      (s = 0 => x <= 2) & (s = 2 => x <= 3)
endinvariant
...
endmodule
• Models a simple probabilistic communication protocol

module ptaexample
    s : [0..2] init 0;
    x : clock;

    invariant
        (s = 0 => x <= 2) &
        (s = 2 => x <= 3)
    endinvariant

    [send] s = 0 & x >= 1 -> 0.9: (s' = 1) & (x' = 0)
           + 0.1: (s' = 2) & (x' = 0);

    [retry] s = 2 & x >= 2 -> 0.95: (s' = 1)
           + 0.05: (s' = 2) & (x' = 0);

endmodule
• **A state of a PTA is a pair** \((l,v) \in \text{Loc} \times \mathbb{R}^X\) **such that** \(v \triangleright inv(l)\)

• **A PTAs start in the initial location with all clocks set to zero**
  – **let** \(0\) **denote the clock valuation where all clocks have value** 0

• **For any state** \((l,v)\), **there is nondeterministic choice between making a discrete transition and letting time pass**
  – **discrete transition** \((l,g,a,p)\) **enabled if** \(v \triangleright g\) **and probability of moving to location** \(l'\) **and resetting the clocks** \(Y\) **equals** \(p(l',Y)\)
  – **time transition** **available only if invariant** \(inv(l)\) **is continuously satisfied while time elapses**
PTA – Example

PTA:

Example execution:

(di,x=0) -> 1.1
(di,x=1.1) ->

(send x≥1 0.9) ->

(retry x≥2 0.95) ->

(x:=0 0.1) ->

(x:=0 0.05) ->

(x≤2 x:=0) ->

(x≥3 x:=0) ->

true x:=0
PTAs – Formal semantics

• Formally, the semantics of a PTA P is an infinite-state MDP \( M_P = (S_P, s_{\text{init}}, \text{Steps}, L_P) \) with:

• States: \( S_P = \{ (l,v) \in \text{Loc} \times \mathbb{R}^X \text{ such that } v \triangleright \text{inv}(l) \} \)

• Initial state: \( s_{\text{init}} = (l_{\text{init}}, 0) \)

• \text{Steps}: \( S_P \rightarrow (\text{Act} \cup \mathbb{R}) \times \text{Dist}(S) \) such that \( (\alpha, \mu) \in \text{Steps}(l,v) \) iff:
  - (time transition) \( \alpha = t \in \mathbb{R}, \mu(l,v+t) = 1 \) and \( v+t' \triangleright \text{inv}(l) \) for all \( t' \leq t \)
  - (discrete transition) \( \alpha = a \in \text{Act} \) and there exists \( (l,g,a,p) \in \text{prob} \)

  such that \( v \triangleright g \) and, for any \( (l',v') \in S_P: \mu(l',v') = \sum_{Y \subseteq X \land v[Y:=0]=v'} p(l', Y) \)

• Labelling: \( L_P(l,v) = L(l) \)

actions of MDP \( M_P \) are the actions of PTA \( P \) or real time delays

multiple resets may give same clock valuation
Probabilistic reachability in PTAs

• For simplicity, in this talk we just consider probabilistic reachability, rather than logic-based model checking
  – i.e. min/max probability of reaching a set of target locations
  – can also encode time-bounded reachability (with extra clock)

• Still captures a wide range of properties
  – probabilistic reachability: “with probability at least 0.999, a data packet is correctly delivered”
  – probabilistic invariance: “with probability 0.875 or greater, the system never aborts”
  – probabilistic time-bounded reachability: “with probability 0.01 or less, a data packet is lost within 5 time units”
  – bounded response: “with probability 0.99 or greater, a data packet will always be delivered within 5 time units”
Overview (Part 2)

- Time, clocks and zones
- Probabilistic timed automata (PTAs)
  - definition, examples, semantics, reachability
- Model checking for PTAs
  - digital clocks
  - zone-based approaches
  - forwards reachability
- Probabilistic hybrid automata (PHAs)
  - definition, examples, semantics, extensions
Digital Clocks

• Represent clocks as **bounded integers**
  – PTA becomes a regular MDP

• **Require two restrictions on PTA:**
  – no open clock constraints (i.e. no $c_1 < 3$, $c_2 > 2$)
  – no diagonals (i.e. no $c_1 \leq c_2$)

• **Then the following properties are preserved:**
  – probabilistic reachability (time- and cost-bounded)
  – expected-time / expected-cost reachability

• **Problem: State space explosion**
  – underlying MDP is exponential in number of clocks and max. constants
Zone-based approaches

- Use zones to construct an MDP

- Conventional **symbolic** model checking relies on computing
  - \( \text{post}(S') \) the states that can be reached from a state in \( S' \) in a single step
  - \( \text{pre}(S') \) the states that can reach \( S' \) in a single step

- Extend these operators to include time passage
  - \( \text{dpost}[e](S') \) the states that can be reached from a state in \( S' \) by **traversing the edge** \( e \)
  - \( \text{tpost}(S') \) the states that can be reached from a state in \( S' \) by **letting time elapse**
  - \( \text{pre}[e](S') \) the states that can reach \( S' \) by **traversing the edge** \( e \)
  - \( \text{tpre}(S') \) the states that can reach \( S' \) by **letting time elapse**
Zone-based approaches

- **Symbolic states** \((l, \zeta)\) where
  - \(l \in \text{Loc}\) (location)
  - \(\zeta\) is a zone over PTA clocks and formula clocks

- **t\text{post}(l,\zeta) = (l, \lnot \zeta \land \text{inv}(l))\)**
  - \(\lnot \zeta\) can be reached from \(\zeta\) by letting time pass
  - \(\lnot \zeta \land \text{inv}(l)\) must satisfy the **invariant** of the location \(l\)

- **t\text{pre}(l,\zeta) = (l, \lnot \zeta \land \text{inv}(l))\)**
  - \(\lnot \zeta\) can reach \(\zeta\) by letting time pass
  - \(\lnot \zeta \land \text{inv}(l)\) must satisfy the **invariant** of the location \(l\)
Zone-based approaches

- For an edge \( e = (l, g, a, p, l', Y) \) where
  - \( l \) is the source
  - \( g \) is the guard
  - \( a \) is the action
  - \( l' \) is the target
  - \( Y \) is the clock reset

- \( dpost[e](l, \zeta) = (l', (\zeta \land g)[Y:=0]) \)
  - \( \zeta \land g \) satisfy the guard of the edge
  - \( (\zeta \land g)[Y:=0] \) reset the clocks \( Y \)

- \( dpre[e](l', \zeta') = (l, [Y:=0]\zeta' \land (g \land inv(l))) \)
  - \( [Y:=0]\zeta' \) the clocks \( Y \) were reset
  - \( [Y:=0]\zeta' \land (g \land inv(l)) \) satisfied guard and invariant of \( l \)
Forwards reachability

• Based on the operation $\text{post}[e](l,\zeta) = tpost(dpost[e](l,\zeta))$
  
  - $(l',v') \in \text{post}[e](l,\zeta)$ if there exists $(l,v) \in (l,\zeta)$ such that after traversing edge $e$ and letting time pass one can reach $(l',v')$

• Forwards algorithm (part 1)
  
  - start with initial state $S_F = \{tpost((l_{init},0))\}$ then iterate
  for each symbolic state $(l,\zeta) \in S_F$ and edge $e$
  add $\text{post}[e](l,\zeta)$ to $S_F$
  - until set of symbolic states $S_F$ does not change

• To ensure termination need to take c–closure of each zone encountered (c is the largest constant in the PTA)
Forwards reachability

- **Forwards algorithm (part 2)**
  - construct finite state MDP \((S_F, (l_{init}, 0), \text{Steps}_F, L_F)\)
  
  - states \(S_F\) (returned from first part of the algorithm)
  - \(L_F(l, \zeta) = L(l)\) for all \((l, \zeta) \in S_F\)
  - \(\mu \in \text{Steps}_F(l, \zeta)\) if and only if
    there exists a probabilistic edge \((l, g, a, p)\) of PTA
    such that for any \((l', \zeta') \in Z:\)

\[
\mu(l', \zeta') = \sum \{ | p(l', X) | (l, g, \sigma, p, l', X) \in \text{edges}(p) \land \text{post}[e](l, \zeta) = (l', \zeta') \}
\]

summation over all the edges of \((l, g, a, p)\) such that applying post to \((l, \zeta)\) leads to the symbolic state \((l', \zeta')\)
Forwards reachability – Example

PTA:

\[
\begin{align*}
I_0 & \xrightarrow{0.5} I_1 & \text{x:=0} \\
I_1 & \xrightarrow{0.5} I_2 & \text{y:=0} \\
I_2 & \xrightarrow{0.5} I_3 & \text{x=0 } \land \text{y=0} \\
I_3 & \xrightarrow{0.5} & \text{y:=0} \\
\end{align*}
\]

MDP:

\[
\begin{align*}
(l_0, x \leq y) & \xrightarrow{0.5} (l_0, x = y) \\
(l_0, x = y) & \xrightarrow{0.5} (l_3, x = y) \\
\end{align*}
\]
Forwards reachability – Limitations

- Problem reduced to analysis of finite-state MDP, but...

- Only obtain **upper bounds on maximum probabilities**
  - caused by when edges are combined

- Suppose post[e₁](l,ζ)=(l₁,ζ₁) and post[e₂](l,ζ)=(l₂, ζ₂)
  - where e₁ and e₂ from the same probabilistic edge

- **By definition of post**
  - there exists (l,vᵢ) ∈ (l,ζ) such that a state in (lᵢ, ζᵢ) can be reached by traversing the edge eᵢ and letting time pass

- Problem
  - we combine these transitions but are (l,v₁) and (l,v₂) the same?
  - may **not exist** states in (l,ζ) for which both edges are enabled
Forwards reachability – Example

- Maximum probability of reaching $l_3$ is 0.5 in the PTA
  - for the left branch need to take the first transition when $x=1$
  - for the right branch need to take the first transition when $x=0$
- However, in the forwards reachability graph probability is 1
  - can reach $l_3$ via either branch from $(l_0, x=y)$
Abstraction Refinement

- Distinguish nondeterminism from model and abstraction
  - yields stochastic game instead of MDP
  - provides lower/upper bounds for min/max probabilities

- If the difference ("error") is too great, refine the abstraction
  - split zones
  - a finer partition yields a more precise abstraction
Overview (Part 2)

- Time, clocks and zones
- Probabilistic timed automata (PTAs)
  - definition, examples, semantics, reachability
- Model checking for PTAs
  - digital clocks
  - zone-based approaches
  - forwards reachability
- Probabilistic hybrid automata (PHAs)
  - definition, examples, semantics, extensions
Timed automata do not always suffice

- Probabilistic timed automata do not always suffice
- Systems with complex dynamics
  - e.g. control processes, vehicle dynamics
- Need general continuous variables instead of clocks
  - behaviour over time given by differential equations
Probabilistic hybrid automata (PHAs)

- Probabilistic hybrid automata (PHAs) extend PTAs by complex dynamics in locations

- Syntax: A PHA is a tuple \((\text{Loc}, \text{l}_{\text{init}}, \text{Act}, \text{X}, \text{inv}, \text{prob}, \text{L})\)
  - \(\text{Loc}\) is a finite set of locations
  - \(\text{l}_{\text{init}} \in \text{Loc} \times \mathbb{R}^\text{X}\) is the initial condition
  - \(\text{Act}\) is a finite set of actions
  - \(\text{X}\) is a finite set of continuous variables
  - \(\text{inv} : \text{Loc} \to 2^{\mathbb{R}^\text{X}}\) is the invariant condition
  - \(\text{flow} : (\text{Loc} \times \mathbb{R}^\text{X}) \to \mathbb{R}^\text{X}\) is the flow condition
  - \(\text{prob} \subseteq \text{Loc} \times 2^{\mathbb{R}^\text{X}} \times \text{Dist}(\text{Loc} \times \mathbb{R}^\text{X})\) is the probabilistic edge relation
  - \(\text{L} : \text{Loc} \to \text{AP}\) is a labelling function

- More general definitions possible

\[\begin{align*}
\text{He} & : T = 2 \\
\text{Co} & : \dot{T} = -T, T \geq 5 \\
\text{Ch} & : T = -T/2, t \leq 1 \\
\text{Er} & : t := 0, t \geq 0.5
\end{align*}\]
Probabilistic edge relation

- **Probabilistic edge relation**
  - \( \text{prob} \subseteq \text{Loc} \times 2^{\mathbb{R}^X} \times \text{Act} \times \text{Dist}(\text{Loc} \times \mathbb{R}^X) \)

- **Probabilistic edge** \((l, g, a, p) \in \text{prob}\)
  - \(l\) is the **source location**
  - \(g\) is the **guard**
  - \(a\) is the **action**
  - \(p\) target **distribution**

- **Edge** \((l, g, a, p, l', Y)\)
  - from probabilistic edge \((l, g, a, p)\) where \(p(l', Y) > 0\)
  - \(l'\) is the **target location**
  - \(Y\) is the **assignment of continuous variables**
Models a simple temperature control

- starts in location \text{He(at)};
- changes between \text{He(at)} and \text{Co(ol)} to adjust temperature
- occasionally moves to \text{Ch(cek)}, where
  - with probability 0.95 can continue its operation
  - with probability 0.05 an \text{Er(ror)} occurs

\text{PHA – Example}
PHAs – Behaviour

• A state of a PTA is a pair \((l,v) \in \text{Loc} \times \mathbb{R}^x\) such that \(v \in \text{inv}(l)\)

• A PTAs start in the initial location with variable assignment given by initial condition

• For any state \((l,v)\), there is a nondeterministic choice between making a discrete transition and letting time pass
  – discrete transition \((l,g,a,p)\) enabled if \(v \triangleright g\) and probability of moving to location \(l'\) and setting variables to \(v'\) equals \(p(l',v')\)
  – time transition available only if invariant \(\text{inv}(l)\) is continuously satisfied while time elapses and the derivate of the trajectory of continuous variables satisfies the invariant
PHA - Example

**Example execution:**

- (He, t=0, T=9) 0.5
- (He, t=0.5, T=10) 1
- (Co, t=0, T=10)
- (Ch, t=0.5, T=7.016)
- (He, t=0, T=7.016) (Er, t=0, T=7.016)
- 1.25
- ...

**PHA:**

- **He**
  - $\dot{T} = 2$
  - $T \leq 10$
  - $t \leq 3$
  - $t := 0$
  - $T \geq 9$
  - $\text{chg}$
- **Co**
  - $\dot{T} = -T$
  - $T \geq 5$
  - $t := 0$
  - $T \leq 9$
  - $\text{chg}$
- **Ch**
  - $\dot{T} = -T/2$
  - $t \leq 1$
  - $0.05$
  - $t := 0$
  - $t \geq 0.5$
  - $\text{chk}$
- **Er**
  - $t \geq 0.5$

**Conditions:**

- $t \geq 2$
- $t := 0$
- $T \leq 3$
- $T \geq 5$
- $t \geq 0.5$
- $t := 0$
• Two more extensions to guarded commands:
  – continuous variables (type \texttt{var})
  – derivative operator \texttt{der} for use in invariants

• Continuous variables
  – evolve over time according to constraints in invariants
  – on transitions, take any value from their domain nondeterministically unless explicitly assigned to

\begin{verbatim}
T : var init 9;

invariant
  T <= 10 \& der(T) = -0.5 * T
endinvariant

[chg] T >= 9 -> (T' = T);
\end{verbatim}
module thermostat

s : [0..3] init 0;
t : var init 0;
T : var init 9;

invariant

(s = 0 => (der(t) = 1 & der(T) = 2 & T <= 10 & t <= 3))
& (s = 1 => (der(t) = 1 & der(T) = -T & T >= 0))
& (s = 2 => (der(t) = 1 & der(T) = -0.5 * T & t <= 1))
& (s = 3 => (der(t) = 0 & der(T) = 0))

endinvariant

[chg] s = 0 & T >= 9  ->  (s' = 1) & (t' = 0) & (T' = T);
[chg] s = 0 & t >= 2  ->  (s' = 2) & (t' = 0) & (T' = T);
[chg] s = 1 & T <= 6  ->  (s' = 0) & (t' = 0) & (T' = T);
[chk] s = 2 & t >= 0.5 ->
    0.95: (s' = 0) & (t' = 0) & (T' = T);
    + 0.05: (s' = 3) & (t' = 0) & (T' = T);

endmodule
PHAs – Formal semantics

- Semantics of PHA \( P \) MDP \( M_P = (S_P, s_{init}, \text{Steps}, L_P) \) with:

- **States**: \( S_P = \{ (l,v) \in \text{Loc} \times \mathbb{R}^X \text{ such that } v \in \text{inv}(l) \} \)

- **Initial state**: \( s_{init} = l_{init} \)

- **Steps**: \( S_P \rightarrow 2^{(\text{Act} \cup \mathbb{R}) \times \text{Dist}(S)} \) such that \( (\alpha, \mu) \in \text{Steps}(l,v) \) iff:
  - **(time transition)** \( \alpha = t \in \mathbb{R} \),
    ex. differentiable flow \( r: [0,t] \rightarrow \mathbb{R}^X \) with \( r(0)=v \), \( r(t') \in \text{inv}(l) \),
    \( \dot{r}(t') \in \text{flow}(l,r(t')) \) for all \( t' \leq t \) and \( \mu(l,r(t))=1 \)
  - **(discrete transition)** \( \alpha = a \in \text{Act} \) and there exists \( (l,g,a,p) \in \text{prob} \)
    such that \( v \in g \) and, for any \( (l',v') \in S_P \): \( \mu(l',v') = p(l',v) \)

- **Labelling**: \( L_P(l,v) = L(l) \)
Deciding properties of PHAs

- Problem: even for nonprobabilistic hybrid automata, reachability is **undecidable**
- Solutions in some cases using **overapproximation**:
  - As for PTAs, subsume concrete states to abstract states
  - Cannot represent exact behaviour in abstraction
  - Rather, build abstraction which simulates the semantics
    - for each step which the semantics can perform, the abstraction has a corresponding step
- Provides **upper bound for maximal reachability**
Abstraction methods for PHAs

- Abstract states: set of finite states

- Have $A \rightarrow B$ if there is $a \in A$ and $b \in B$ so that $a \rightarrow b$

- Similar for probabilistic case by summing up probabilities

- Construct abstractions for PHAs by adapting existing methods for nonprobabilistic HAs
Abstraction methods for PHAs

- Different abstraction methods: e.g. using rectangles
- Start with bounded variable space
- Divide into rectangles
- Check which ones are connected
- To refine: split rectangles, or disprove paths

\[\text{Probabilistic jump}\]

\[\text{Flow}\]

\[0.3\]

\[0.1\]

\[0.6\]

\[e.g. \ [RS07]\]
Abstraction methods for PHAs

- Other methods based on polyhedra [HH94,Frehse05]
- Forward or backward reachability analysis
- Enclose flows by polyhedra
- Jumps similar to PTA
- To refine: decrease split length
Abstraction methods for PHAs

- Other methods based on predicates [ADI06b]
- Fix finite set of predicates over variables
  - E.g. \{Loc=\text{Heat} \land T \leq t, \text{Loc=Check} \land t=T-2, \ldots \}\n- Each abstract state assigns truth value to each predicate
- Transitions can then be decided by Satisfiability Modulo Theories (SMT)
- Refinement: introduce new predicates
  - similar to non-hybrid predicate abstraction

\[ p_1 = (\text{Loc=Heat} \land T \leq t) \]
\[ p_2 = (\text{Check} \land t=T-2) \]
\[ p_3 = (\ldots) \]
Continuous nondeterminism

- Can extend to **continuous** nondeterminism in jumps
- And to **differential** inequations

- **He**
  - $\dot{T} = 2$
  - $1 \leq T \leq 2$
  - $T \leq 10$
  - $t \leq 3$

- **Co**
  - $\dot{T} = -T$
  - $T \geq 5$

- **Ch**
  - $\dot{T} = -T/2$
  - $T \leq 1$

- **Er**

On update, $t$ can become any value between 0 and 1

Derivative of $T$ is any value between 1 and 2
Continuous distributions

- Often continuous probability distributions of interest
  - Measurements (normal distribution)
  - Random delays (exponential distribution)

\[ \text{density of normal distribution} \]
\[ \text{can state probability of certain set of successors} \]
\[ \text{but individual successors have probability 0} \]
Well-definedness

- Semantics: nondeterministic Markov process (NLMP)
- no PA, because of issues with measurability
- restrictions on transitions necessary

\[ x := 0 \quad \text{true} \quad x \in V \]

- \( M \): normal distribution
- \( V \): some Vitali set
- Probability to reach rightmost mode?
- Does not exist!
- Because \( V \) is not measurable
- Thus: restrictions on automaton components necessary
- Carries over to well-definedness of semantics
Solution methods

- Solution methods no longer apply directly
  - divide continuous support into fixed number of parts
- Afterwards, can apply methods discussed for PHAs
Rewards

- So far, considered only reachability
- Extension to reward-based properties possible
- Extend simulation relation to take reward into account
  - basically, reward in abstraction higher than in semantics
- Extend abstraction by reward structure
- Can analyse similar properties as for basic MDPs
  - cumulative, long-run, etc.

\[\text{semantics} \simulates \text{abstraction}\]

[HH13]
Game-based Abstraction

- Also game-based abstraction possible
- Allows also to bound reachability probability from both below and above

\[ p_{s}^\text{min} \quad p_{s}^\text{max} \]

- Using similar methods as in the PTA case

[HNP+11]
Other notions of PHAs

- Many other notions of PHAs exist
- All of them have some discrete–continuous features
- But all with different behaviours and definitions

<table>
<thead>
<tr>
<th></th>
<th>SHA</th>
<th>STA</th>
<th>PHA</th>
<th>PPTA</th>
<th>PTA</th>
<th>PA</th>
<th>SHS</th>
<th>DTS</th>
<th>SHS</th>
<th>NLMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>discrete stochastics</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>continuous stochastics</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>discrete dynamics</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>real time</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>differential inclusions</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>stochastic differential eqs.</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>discrete nondeterminism</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>continuous nondeterminism</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
</tr>
</tbody>
</table>
Piecewise-deterministic Markov processes

- No nondeterminism in basic notion (extensions exist)
- But rate-driven jumps
- In each mode have vector field for continuous behaviour
- Jumps occur when border of mode hit
- Or according to a certain rate
  - Which may depend on time and valuation of variables
Stochastic Hybrid Systems by Hu et al

- No nondeterminism

- But stochastic differential equations within modes
  \[ dX(t) = f(Q(\tau_n), X(t)) \, dt + g(Q(\tau_n), X(t)) \, dB_t \]

- Different solution methods exist for different properties
  - E.g. time-discretisation

[HLS00]
Discrete-time SHS by Abate et al

- **State**: mode + evaluation of continuous variables
- **Discrete-time model**
  - E.g. by time-discretisation of SHS
- each step choose successor mode and variable valuation

Solution method: discretisation
Divide to finitely many regions, transform to Markov model

[APLS08]
• Basic idea for PTAs
  – reduce to the analysis of a finite-state model
  – in most cases, this is a Markov decision process (MDP)

• Approaches:
  – digital clocks [KNPS06]
  – forwards reachability [KNSS02]
  – game-based abstraction refinement [KNP09c]

• For PHAs
  – more general behaviours possible than in PTAs
  – can not reduce to equivalent finite model (undecidability)
  – can compute overapproximation
  – a number of abstraction methods exist
  – continuous distributions, rewards, game-based abstraction

• A number of related approaches exists