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

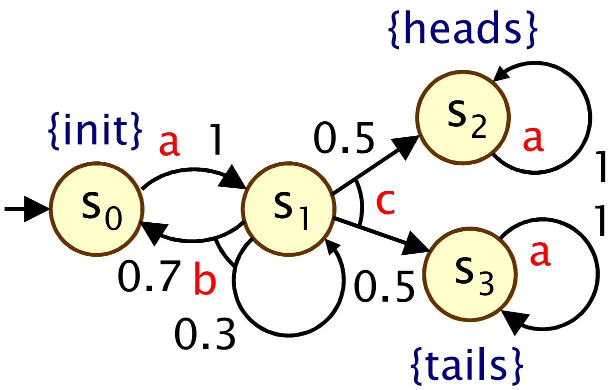
CPSWeek'13, Philadelphia, April 2013

Part 2

Probabilistic Hybrid Systems

Recall – MDPs

- 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. $P_{<0.1}[\Diamond \mathrm{err}]$ – 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

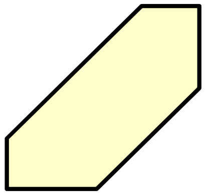
Time, clocks and clock valuations

- 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
    - $(v+t)(x) = v(x)+t \quad \forall x \in X$
  - $v[Y:=0]$: 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 Zones(X):

  $\zeta ::= x \leq d \;|\; c \leq x \;|\; x+c \leq y+d \;|\; \neg\zeta \;|\; \zeta \vee \zeta$



  - 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. $\zeta_1 \wedge \zeta_2 \equiv \neg(\neg\zeta_1 \vee \neg\zeta_2)$
  - 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

Zones and clock valuations

- A clock valuation v satisfies a zone ζ, written $v \triangleright \zeta$ if
  - ζ resolves to true after substituting each clock x with $v(x)$

- The semantics of a zone $\zeta \in Zones(X)$ 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. $(x \leq 2) \wedge (y \leq 1) \wedge (x \leq y+2)$ and $(x \leq 2) \wedge (y \leq 1) \wedge (x \leq y+3)$

- 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

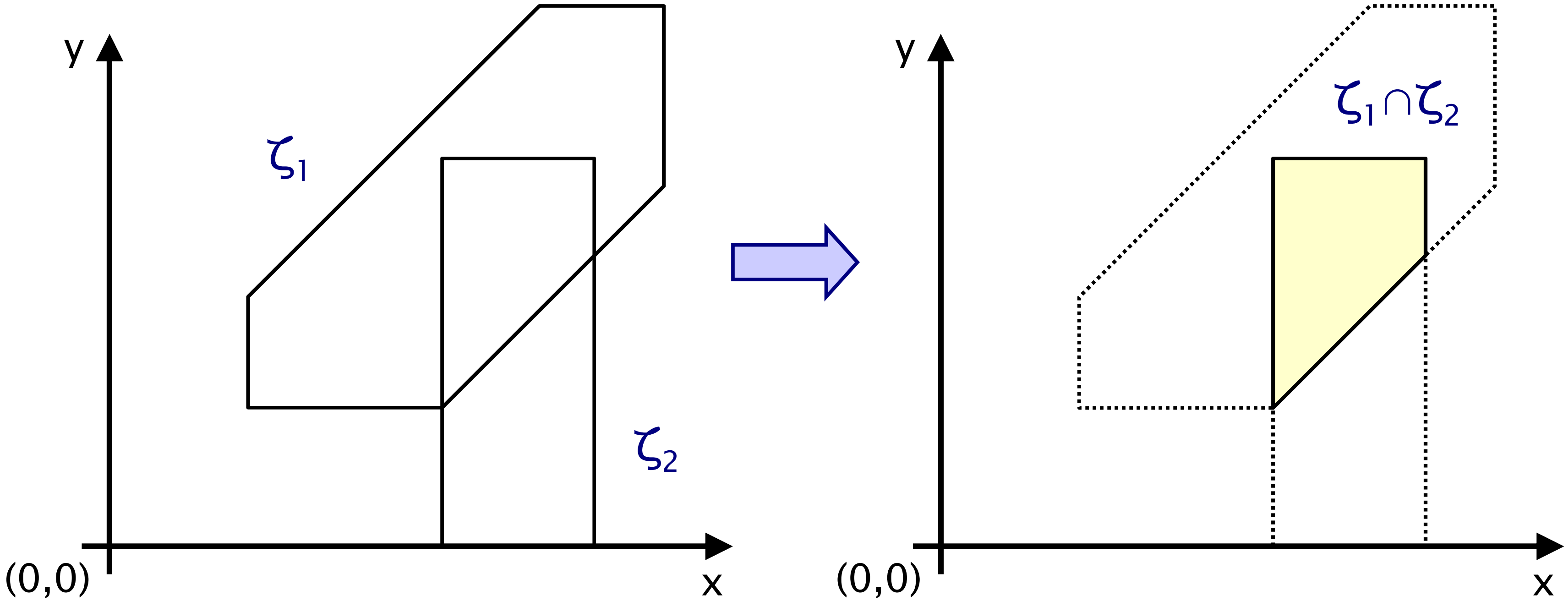
c-equivalence and c-closure

- 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 ζ, denoted close(ζ,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 ζ if the clocks in Y are reset: $[Y{:=}0]\zeta$
  - Forwards diagonal projection: $\nearrow \zeta$

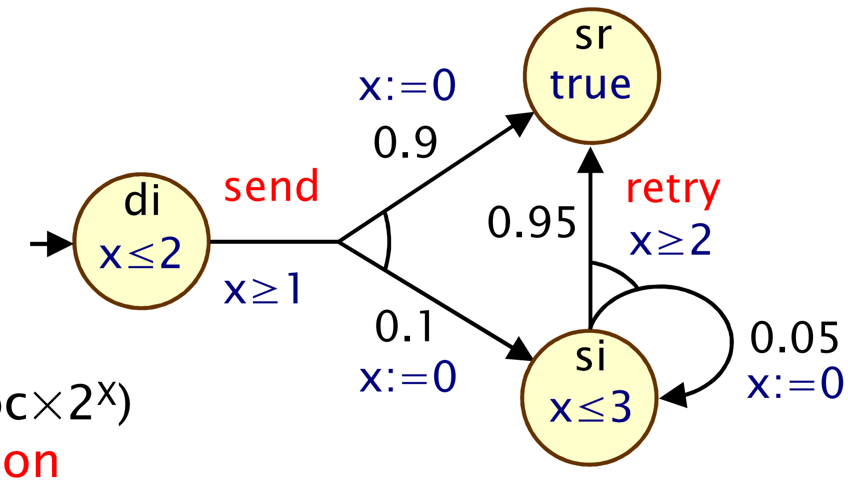
Overview (Part 2)

- 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

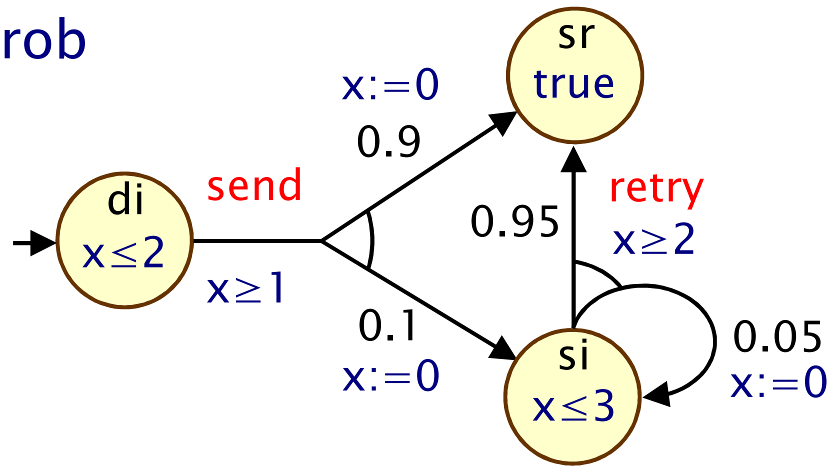
- Syntax: A PTA is a tuple (Loc, l_{init}, Act, X, inv, prob, L)
  - Loc is a finite set of locations
  - $l_{init} \in$ Loc is the initial location
  - Act is a finite set of actions
  - X is a finite set of clocks
  - inv : Loc → Zones(X) is the invariant condition
  - prob ⊆ Loc×Zones(X)×Dist(Loc×2^X) is the probabilistic edge relation
  - L : Loc → AP is a labelling function



Probabilistic edge relation

- Probabilistic edge relation
  - prob $\subseteq$ Loc$\times$Zones(X)$\times$Act$\times$Dist(Loc$\times 2^X$)

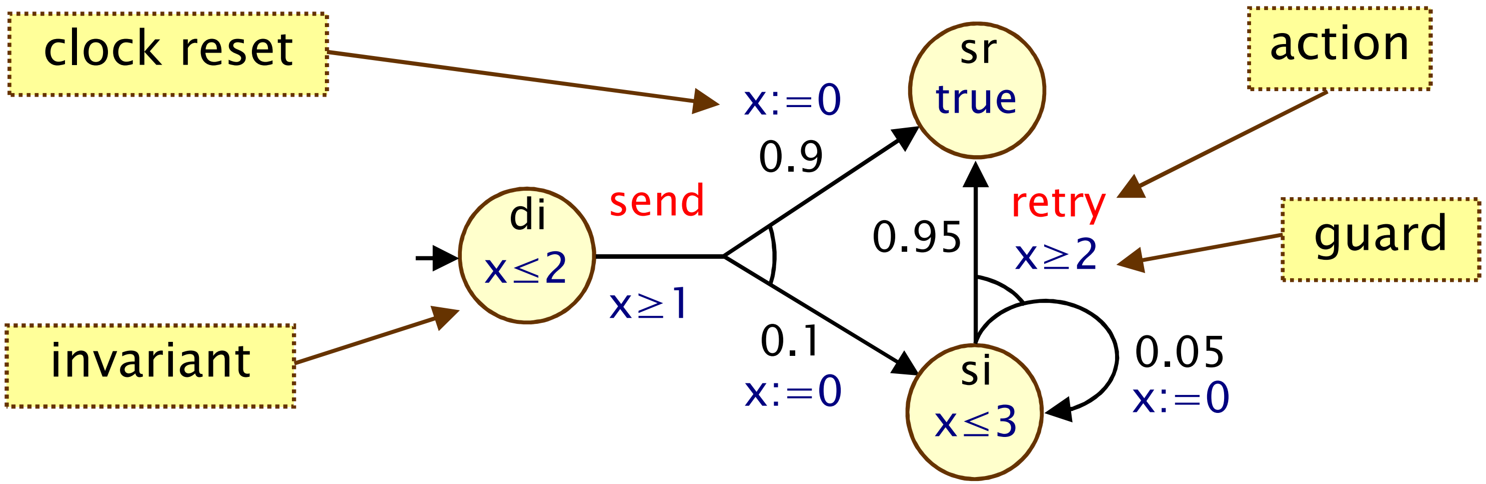
- Probabilistic edge (l,g,a,p) $\in$ prob
  - l is the source location
  - g is the guard
  - a is the action
  - p target distribution



- 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

PTA – Example

- 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

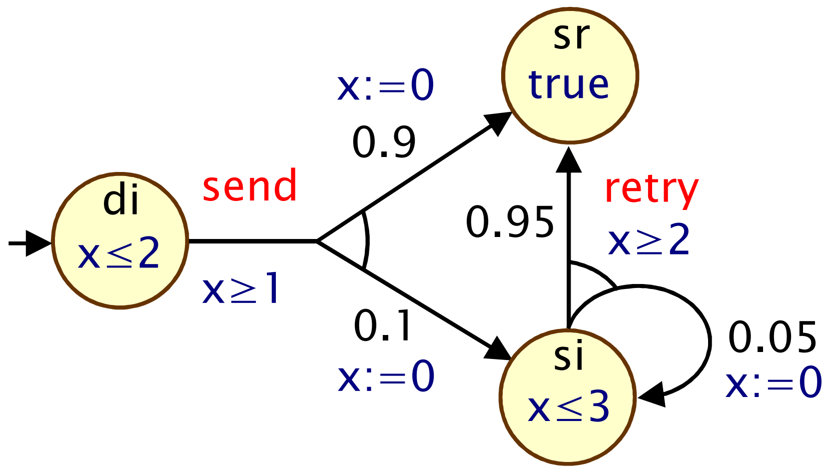
- Simple extension of guarded commands:
  - new variable type clock
  - new language construct invariant

- 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
```

PTA – Example

- 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
```

PTAs - Behaviour

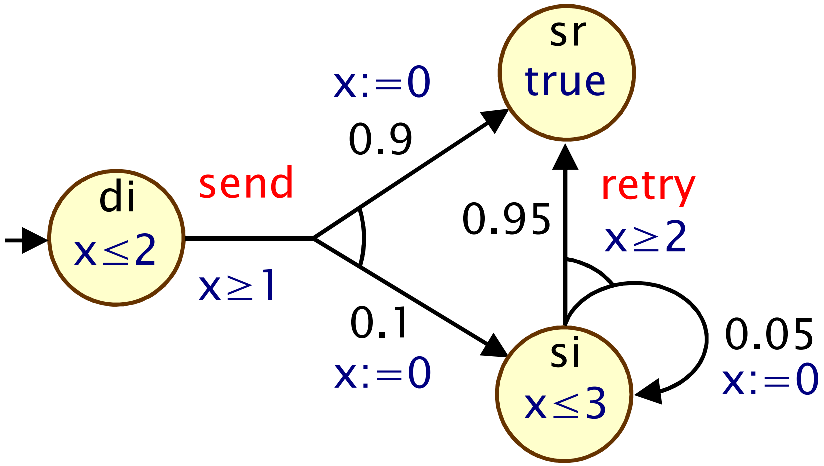
- A state of a PTA is a pair $(l,v) \in Loc \times \mathbb{R}^X$ such that $v \triangleright inv(l)$

- A PTAs start in the initial location with all clocks set to zero
  - let $\underline{0}$ denote the clock valuation where all clocks have value 0

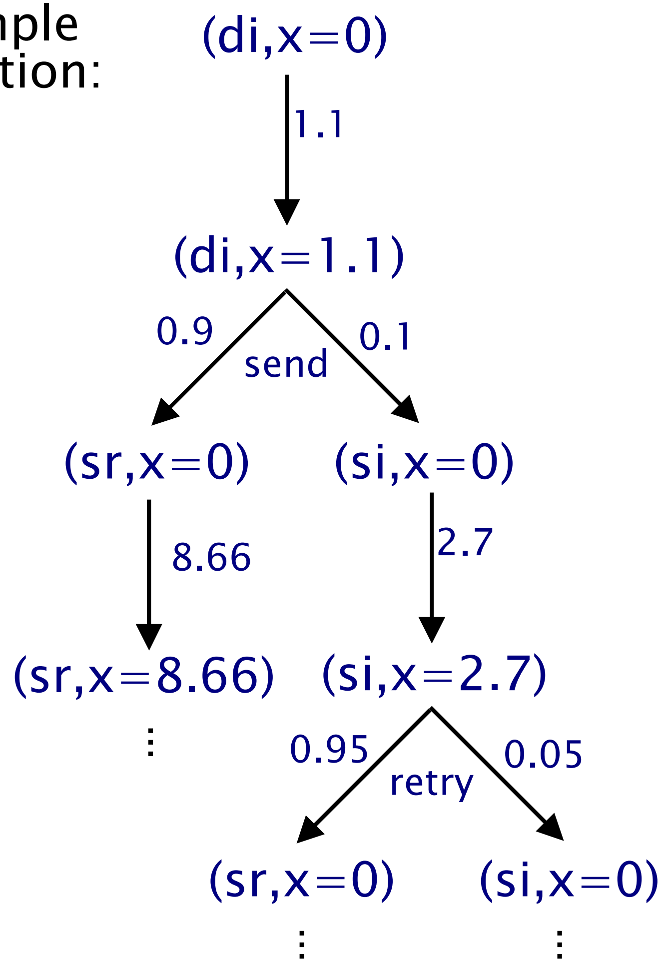
- For any state (l,v), there is nondeterministic choice between making a discrete transition and letting time pass
  - discrete transition (l,g,a,p) enabled if $v \triangleright g$ and probability of moving to location l' and resetting the clocks Y equals $p(l',Y)$
  - time transition available only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

PTA – Example

PTA:



Example execution:



PTAs – Formal semantics

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

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

- Initial state: $s_{init} = (l_{init}, \underline{0})$

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

- **Steps**: $S_P \rightarrow 2^{(Act \cup \mathbb{R}) \times Dist(S)}$ such that $(\alpha, \mu) \in \mathbf{Steps}(l,v)$ iff:
  - (time transition) $\alpha = t \in \mathbb{R}$, $\mu(l,v+t)=1$ and $v+t' \triangleright inv(l)$ for all $t' \leq t$
  - (discrete transition) $\alpha = a \in Act$ and there exists $(l,g,a,p) \in prob$

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

- Labelling: $L_P(l,v) = L(l)$

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
  - **post**(S') the states that can be reached from a state in S' in a single step
  - **pre**(S') the states that can reach S' in a single step

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

Zone-based approaches

- Symbolic states (l, ζ) where
  - $l \in Loc$ (location)
  - ζ is a zone over PTA clocks and formula clocks
- **tpost**$(l,\zeta) = (l, \nearrow\zeta \wedge inv(l))$
  - $\nearrow\zeta$ can be reached from ζ by letting time pass
  - $\nearrow\zeta \wedge inv(l)$ must satisfy the invariant of the location l
- **tpre**$(l,\zeta) = (l, \swarrow\zeta \wedge inv(l))$
  - $\swarrow\zeta$ can reach ζ by letting time pass
  - $\swarrow\zeta \wedge inv(l)$ 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,ζ) = (l', $(\zeta\wedge g)[Y:=0]$)
  - $\zeta\wedge g$ satisfy the guard of the edge
  - $(\zeta\wedge g)[Y:=0]$ reset the clocks Y

- **dpre**[e](l',ζ') = (l, $[Y:=0]\zeta' \wedge (g \wedge inv(l))$)
  - $[Y:=0]\zeta'$ the clocks Y were reset
  - $[Y:=0]\zeta' \wedge (g \wedge inv(l))$ satisfied guard and invariant of l

Forwards reachability

- Based on the operation **post**[e](l,ζ) = **tpost**(**dpost**[e](l,ζ))
  - $(l',v') \in$ **post**[e](l,ζ) 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**($(l_{init},\underline{0})$)} then iterate

    for each symbolic state $(l,\zeta) \in S_F$ and edge e

    add **post**[e](l,ζ) 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 $(S_F,(l_{init},\underline{0}),\mathbf{Steps}_F,L_F)$
  - states S_F (returned from first part of the algorithm)
  - $L_F(l,\zeta)=L(l)$ for all $(l,\zeta) \in S_F$
  - $\mu \in \mathbf{Steps}_F(l,\zeta)$ if and only if

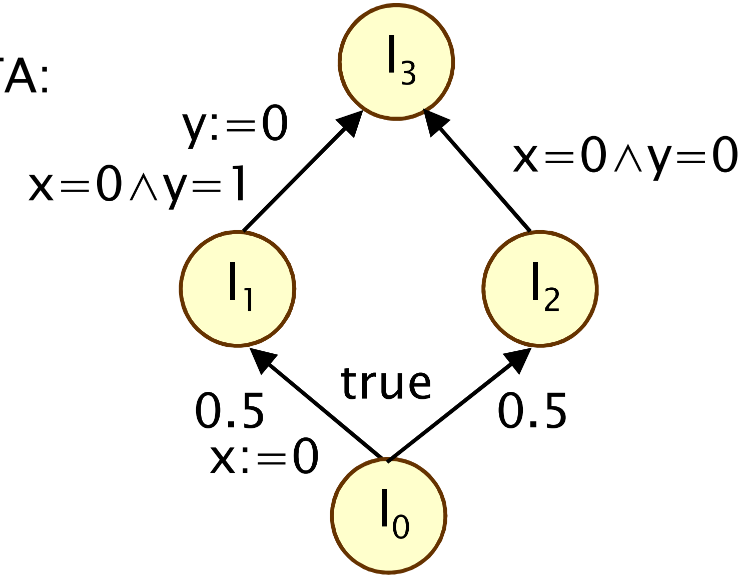
    there exists a probabilistic edge (l,g,a,p) of PTA such that for any $(l', \zeta') \in Z$:

$$\mu(l',\zeta') = \sum \{\!|\, p(l',X) \mid (l,g,\sigma,p,l',X) \in \mathrm{edges}(p) \wedge \mathbf{post}[e](l,\zeta) = (l',\zeta') \,|\!\}$$

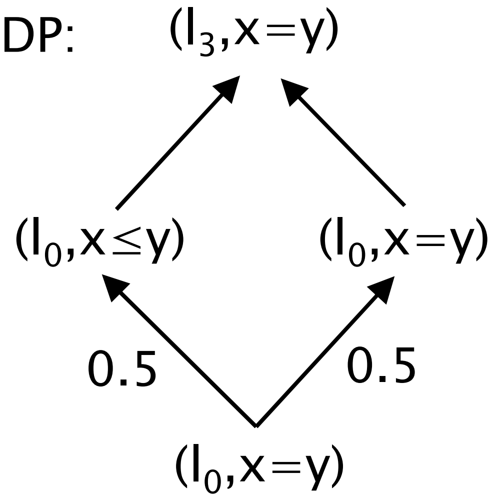
summation over all the edges of (l,g,a,p) such that applying **post** to (l,ζ) leads to the symbolic state (l',ζ')

Forwards reachability – Example

PTA:



MDP:



Forwards reachability – Limitations

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

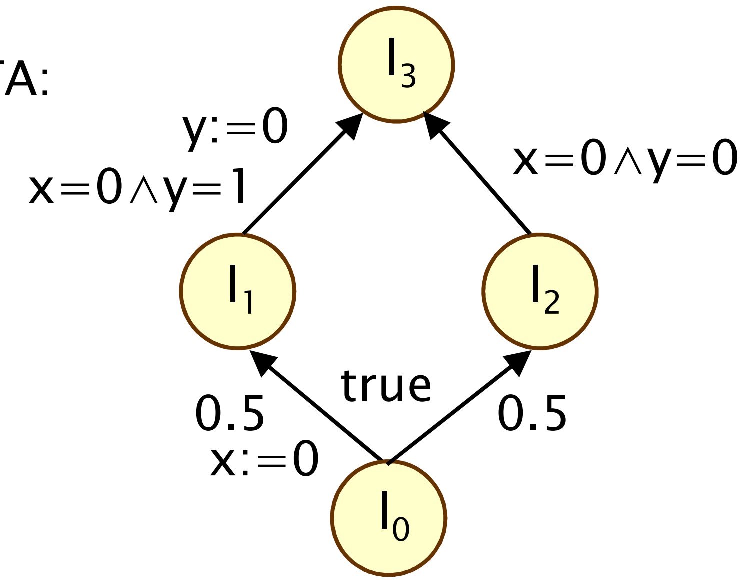
- Only obtain upper bounds on maximum probabilities
  - caused by when edges are combined

- Suppose **post**$[e_1](l,\zeta)=(l_1,\zeta_1)$ and **post**$[e_2](l,\zeta)=(l_2, \zeta_2)$
  - where e_1 and e_2 from the same probabilistic edge
- By definition of **post**
  - there exists $(l,v_i) \in (l,\zeta)$ such that a state in (l_i, ζ_i) can be reached by traversing the edge e_i and letting time pass
- Problem
  - we combine these transitions but are (l,v_1) and (l,v_2) the same?
  - may not exist states in (l,ζ) for which both edges are enabled

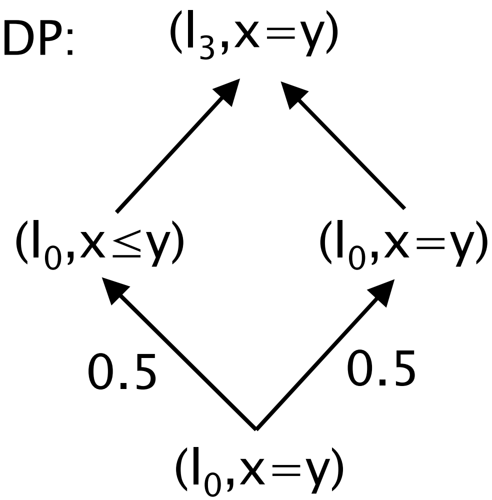
Forwards reachability – Example

- 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 $(l_0,x=y)$

PTA:

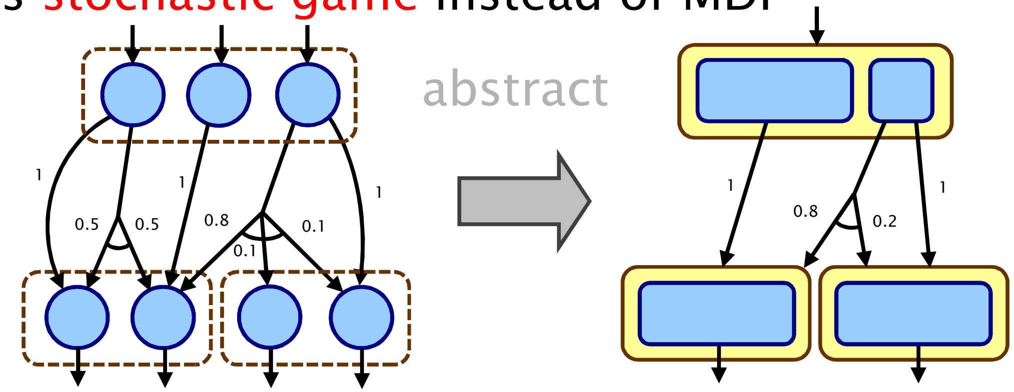


MDP:

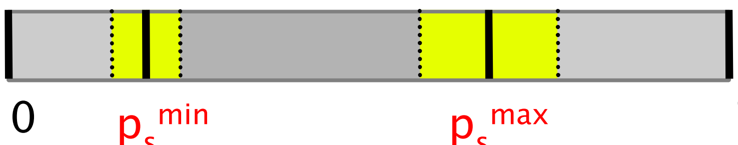


Abstraction Refinement

- Distinguish nondeterminism from model and abstraction
  - yields stochastic game instead of MDP



  - provides lower/upper bounds for min/max probabilities



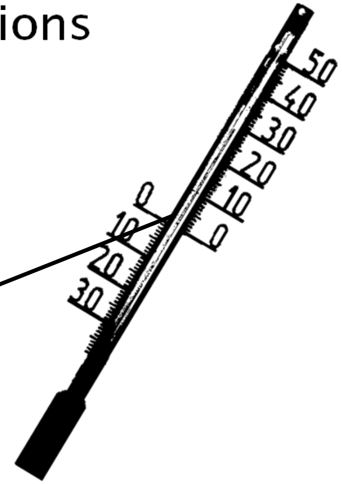
- If the difference ("error") is too great, refine the abstraction
  - split zones
  - a finer partition yields a more precise abstraction

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

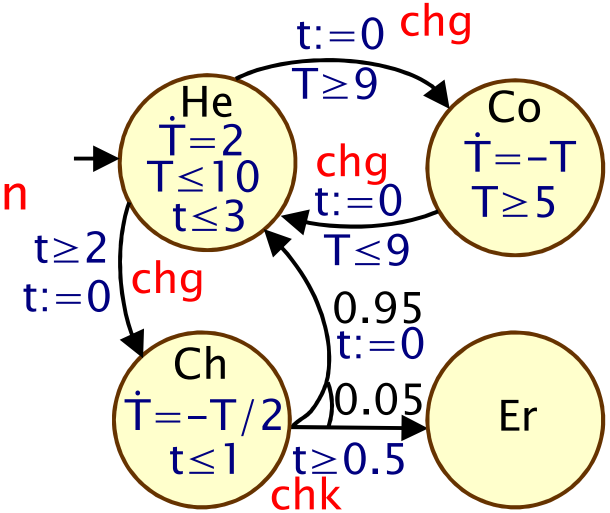
- Probabilistic timed automata do not always suffice
- Systems with complex dynamics
  - e.g. control processes, vehicle dynamics
- Need general continuous variables instead of clocks
  - behaviour over time given by differential equations



Probabilistic hybrid automata (PHAs)

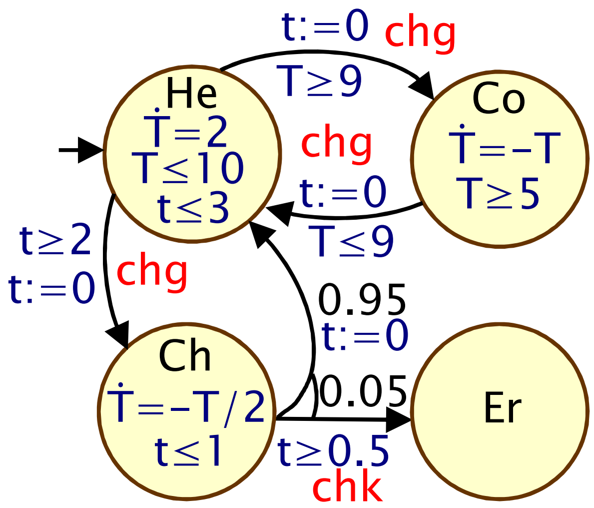
- Probabilistic hybrid automata (PHAs) [Spr01]
  - extend PTAs by complex dynamics in locations

- Syntax: A PHA is a tuple (Loc, l_{init}, Act, X, inv, prob, L)
  - Loc is a finite set of locations
  - $l_{init} \in Loc \times \mathbb{R}^X$ is the initial condition
  - Act is a finite set of actions
  - X is a finite set of continuous variables
  - inv : Loc → $2^{\mathbb{R}^X}$ is the invariant condition
  - flow : (Loc × $\mathbb{R}^X$) → $\mathbb{R}^X$ is the flow condition
  - prob ⊆ $Loc \times 2^{\mathbb{R}^X} \times Dist(Loc \times \mathbb{R}^X)$ is the probabilistic edge relation
  - L : Loc → AP is a labelling function
- More general definitions possible



Probabilistic edge relation

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

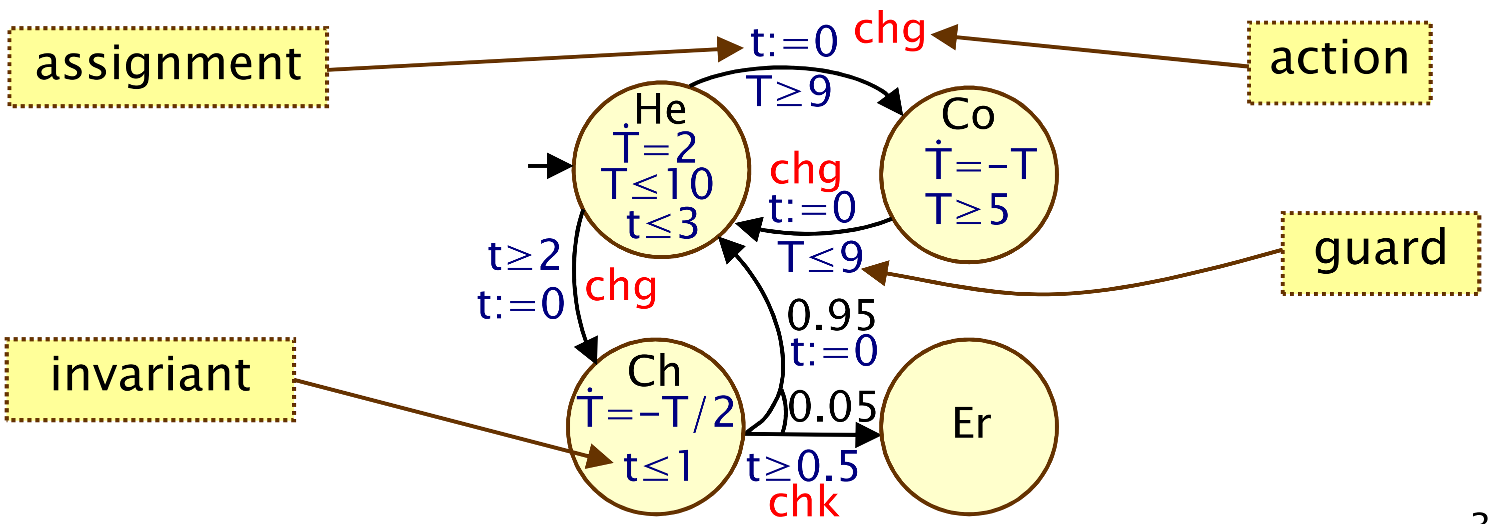


- Probabilistic edge (l,g,a,p) $\in$ prob
  - l is the source location
  - g is the guard
  - a is the action
  - p target distribution

- 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

PHA – Example

- Models a simple temperature control
  - starts in location He(at);
  - changes between He(at) and Co(ol) to adjust temperature
  - occasionally moves to Ch(eck), where
    - with probability 0.95 can continue its operation
    - with probability 0.05 an Er(ror) occurs

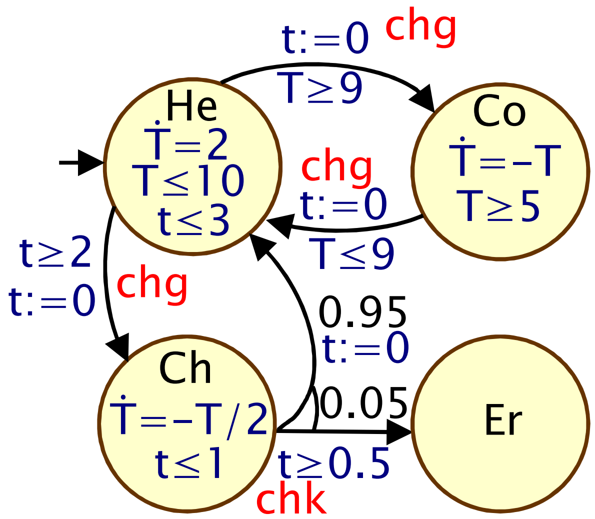


PHAs – Behaviour

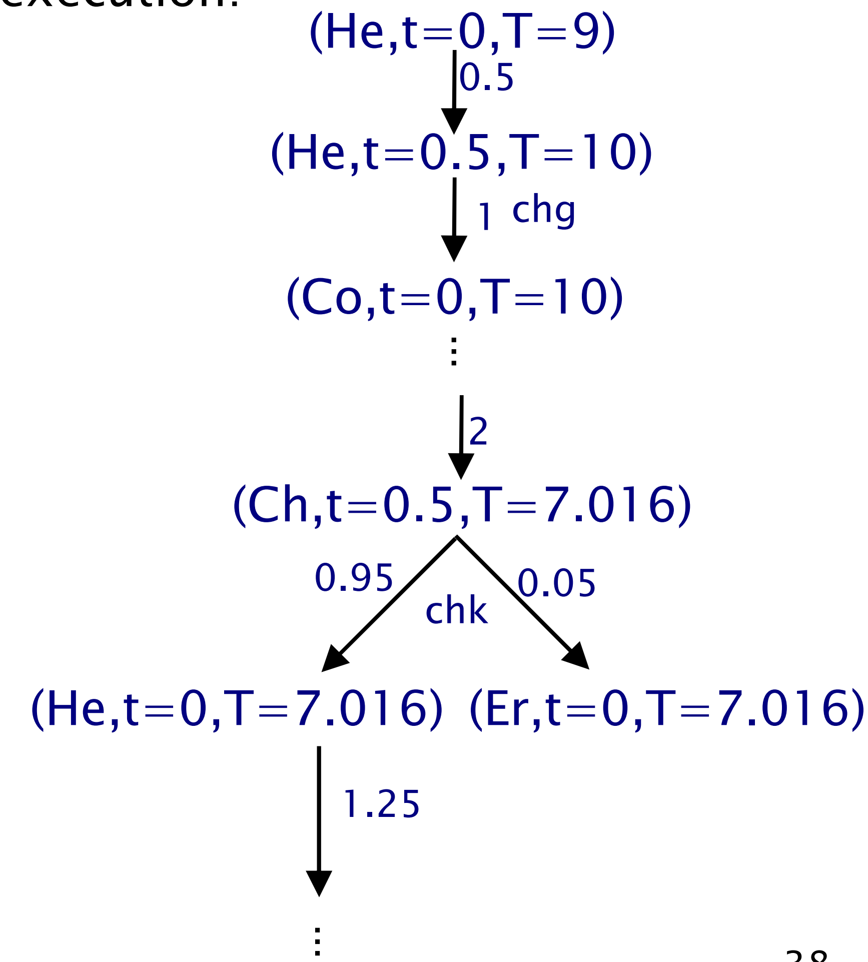
- A state of a PTA is a pair $(l,v) \in Loc \times \mathbb{R}^X$ such that $v \in inv(l)$
- A PTAs start in the initial location with variable assignment given by initial condition
- For any state (l,v), there is a nondeterministic choice between making a discrete transition and letting time pass
  - 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 $inv(l)$ is continuously satisfied while time elapses and the derivate of the trajectory of continuous variables satisfies the invariant

PHA – Example

PHA:



Example execution:



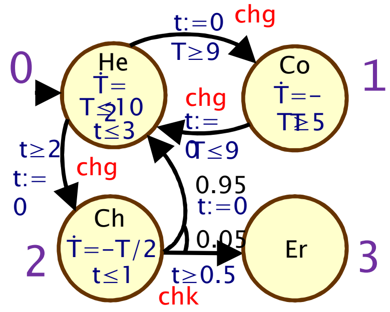
PHA Modelling

- Two more extensions to guarded commands:
  - continuous variables (type var)
  - derivative operator 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

```
T : var init 9;
invariant
    T <= 10 & der(T) = -0.5 * T
endinvariant
[chg] T >= 9 -> (T' = T);
```

PHA - Example




```
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 $M_P = (S_P, s_{init}, \mathbf{Steps}, L_P)$ with:

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

- Initial state: $s_{init} = l_{init}$

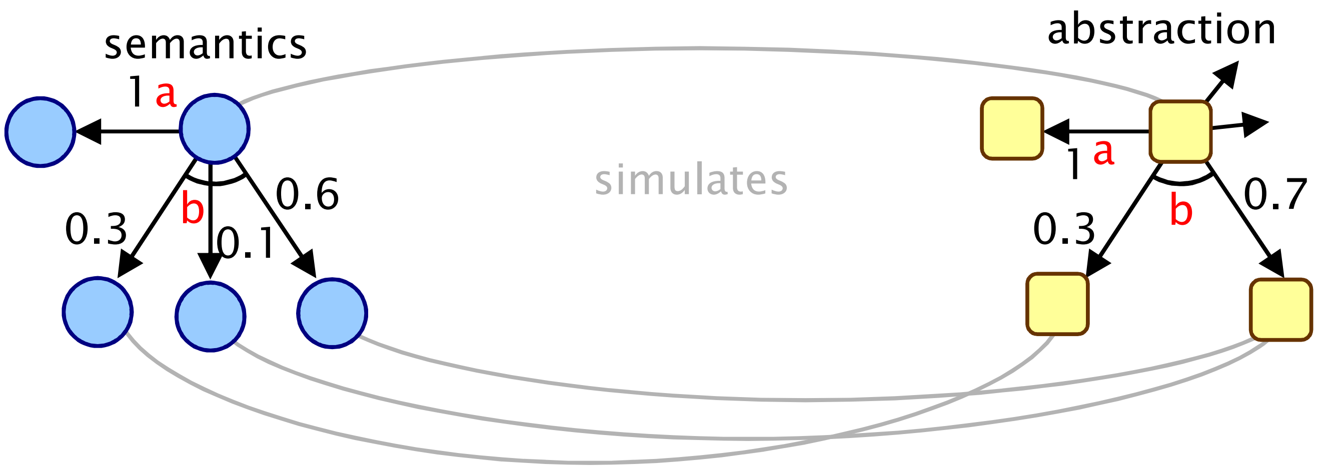
actions of MDP M_P are the actions of PHA P or real time delays

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

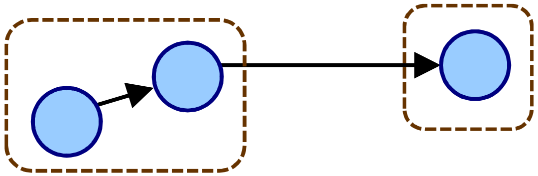
- Problem: even for nonprobabilistic hybrid automata, reachability is undecidable
- Solutions in some cases using overapproximation:
- As for PTAs, subsume concrete states to abstract states
- Cannot represent exact behaviour in abstraction
- Rather, build abstraction which simulates the semantics
 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

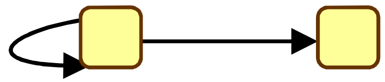


Abstraction methods for PHAs

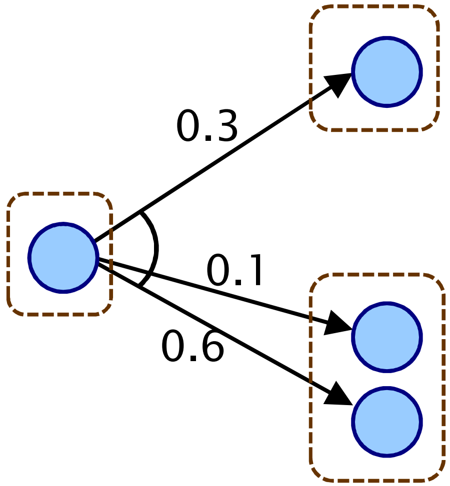
- Abstract states: set of finite states

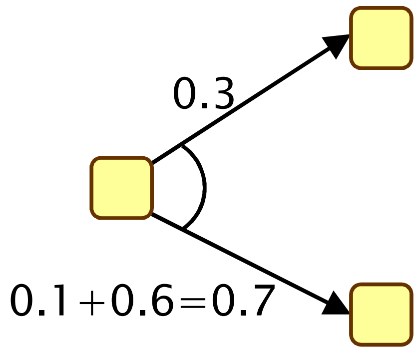
- Have A→B if there is a∈A and b∈B so that a→b





- Similar for probabilistic case by summing up probabilities



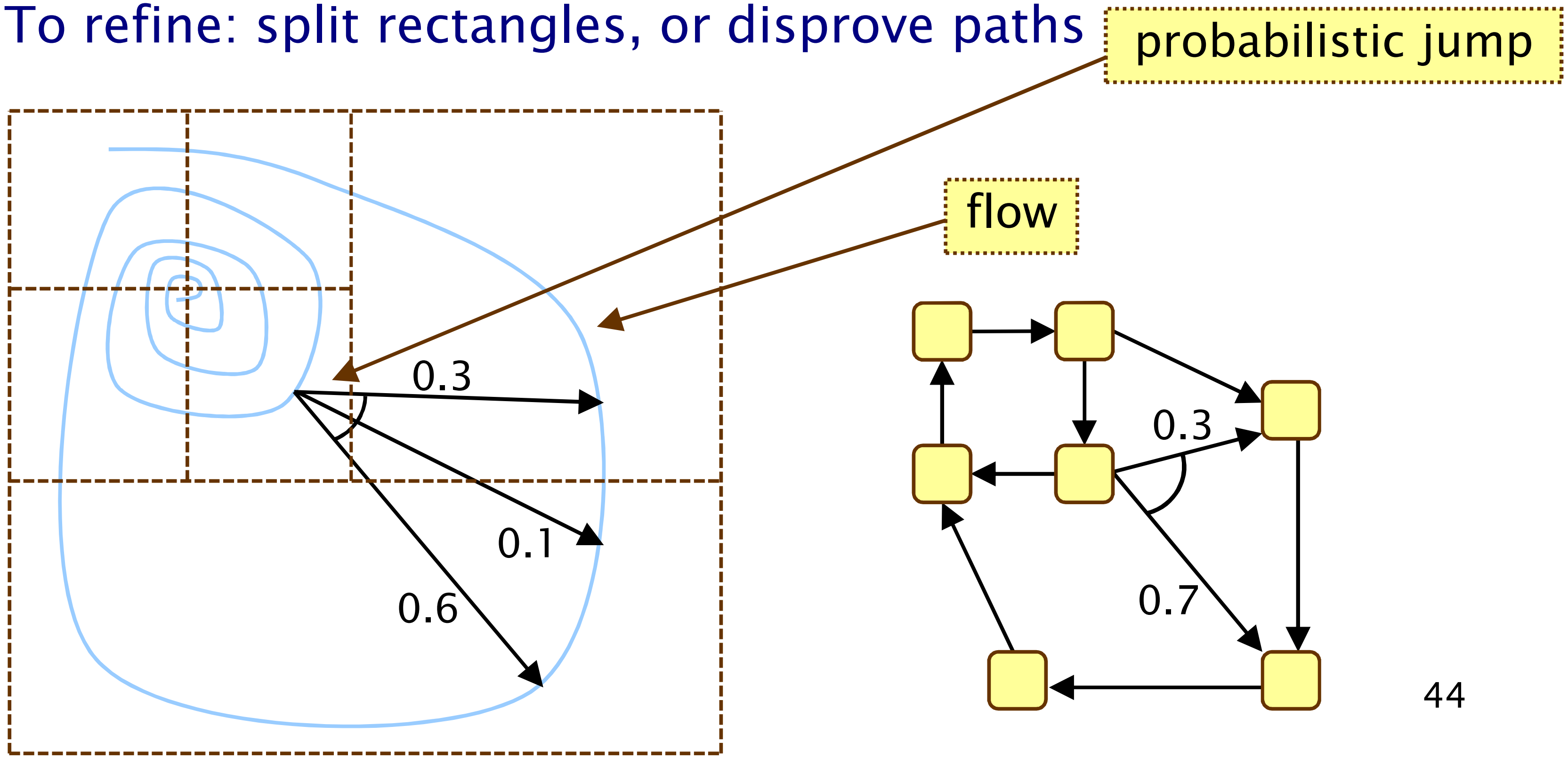


- Construct abstractions for PHAs
  by adapting existing methods for nonprobabilistic HAs

Abstraction methods for PHAs

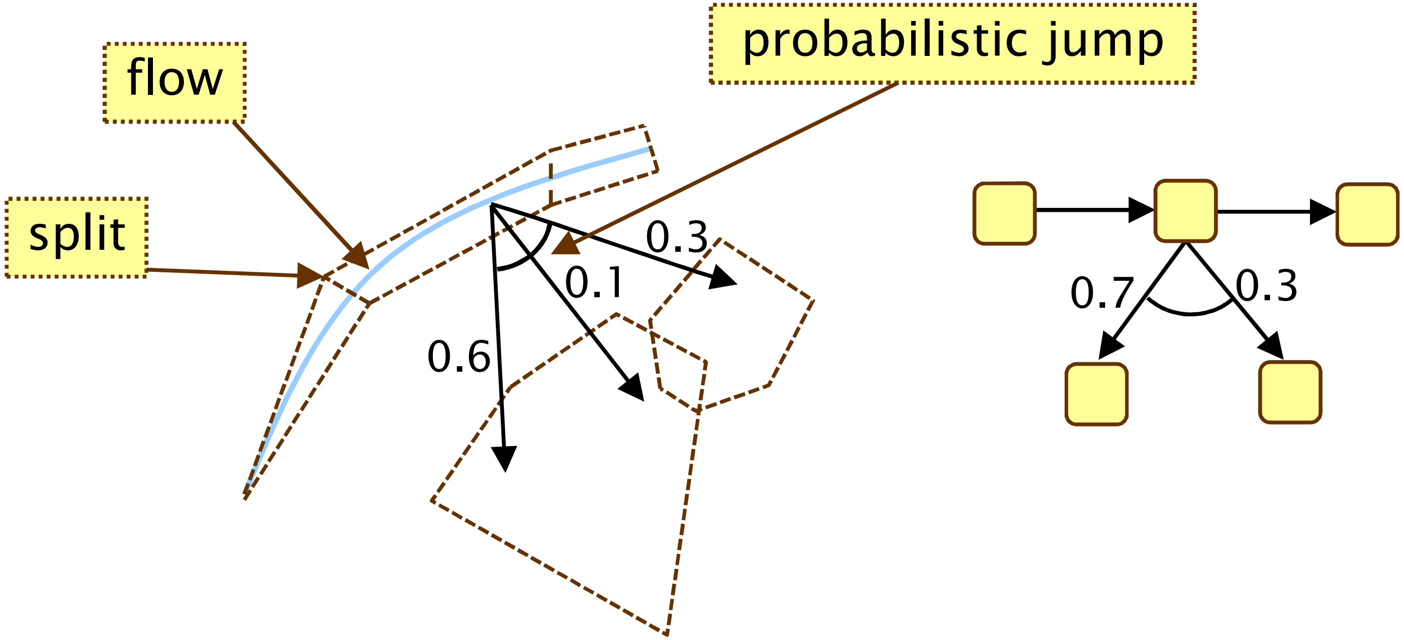
- Different abstraction methods: e.g. using rectangles
- Start with bounded variable space
- Divide into rectangles
- Check which ones are connected
- To refine: split rectangles, or disprove paths

e.g. [RS07]



Abstraction methods for PHAs

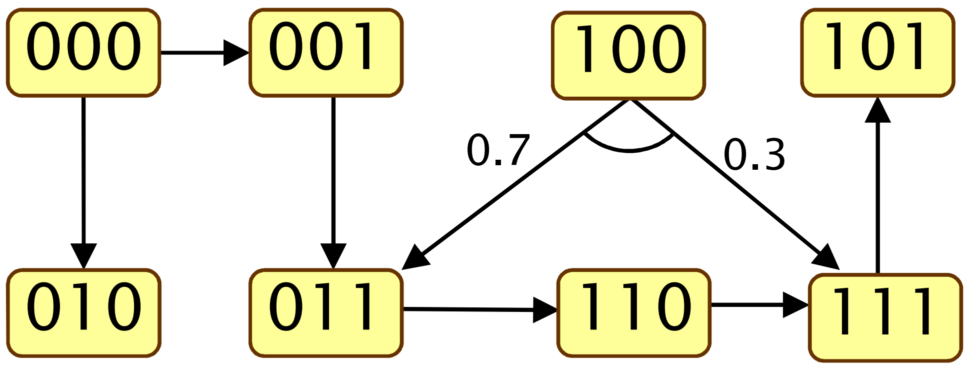
- Other methods based on polyhedra [HH94,Frehse05]
- Forward or backward reachability analysis
- Enclose flows by polyhedra
- Jumps similar to PTA
- To refine: decrease split length



Abstraction methods for PHAs

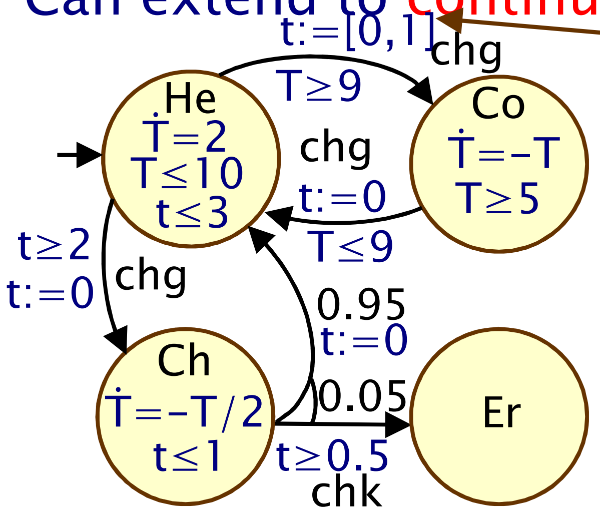
- Other methods based on predicates [ADI06b]
- Fix finite set of predicates over variables
  - E.g. {Loc=Heat$\wedge$T$\leq$t, Loc=Check$\wedge$t=T-2, … }
- Each abstract state assigns truth value to each predicate
- Transitions can then be decided by
  Satisfiability Modulo Theories (SMT)
- Refinement: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

p_1=(Loc=Heat$\wedge$T$\leq$t)
p_2=(Check$\wedge$t=T-2)
p_3=(…)



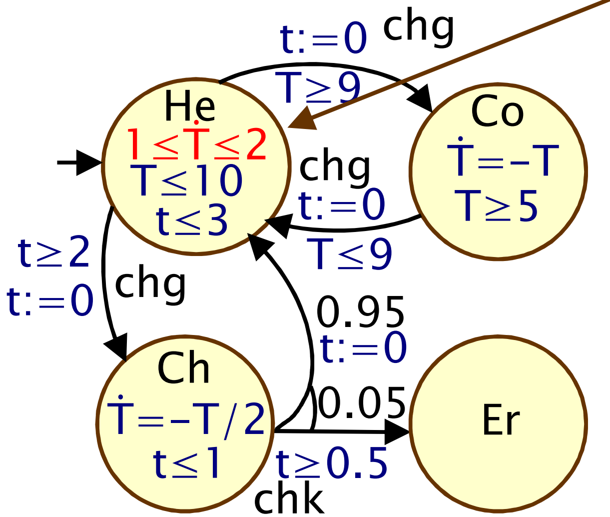
Continuous nondeterminism

- Can extend to continuous nondeterminism in jumps

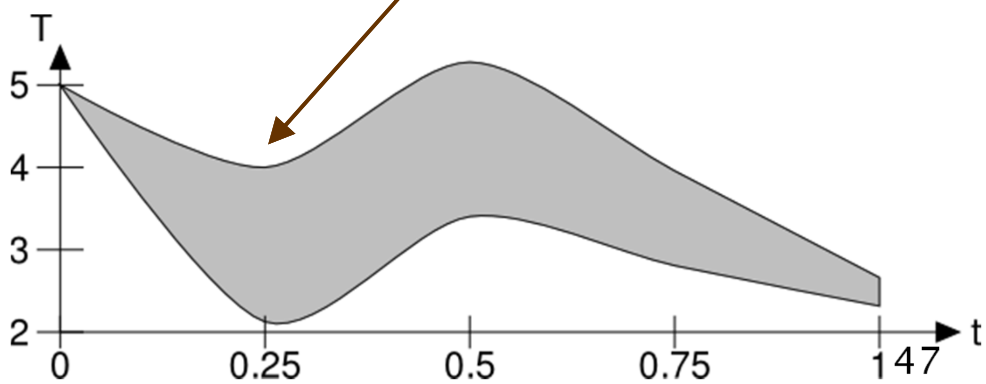


on update, t can become any value between 0 and 1

- And to differential inequations



derivative of T is any value between 1 and 2

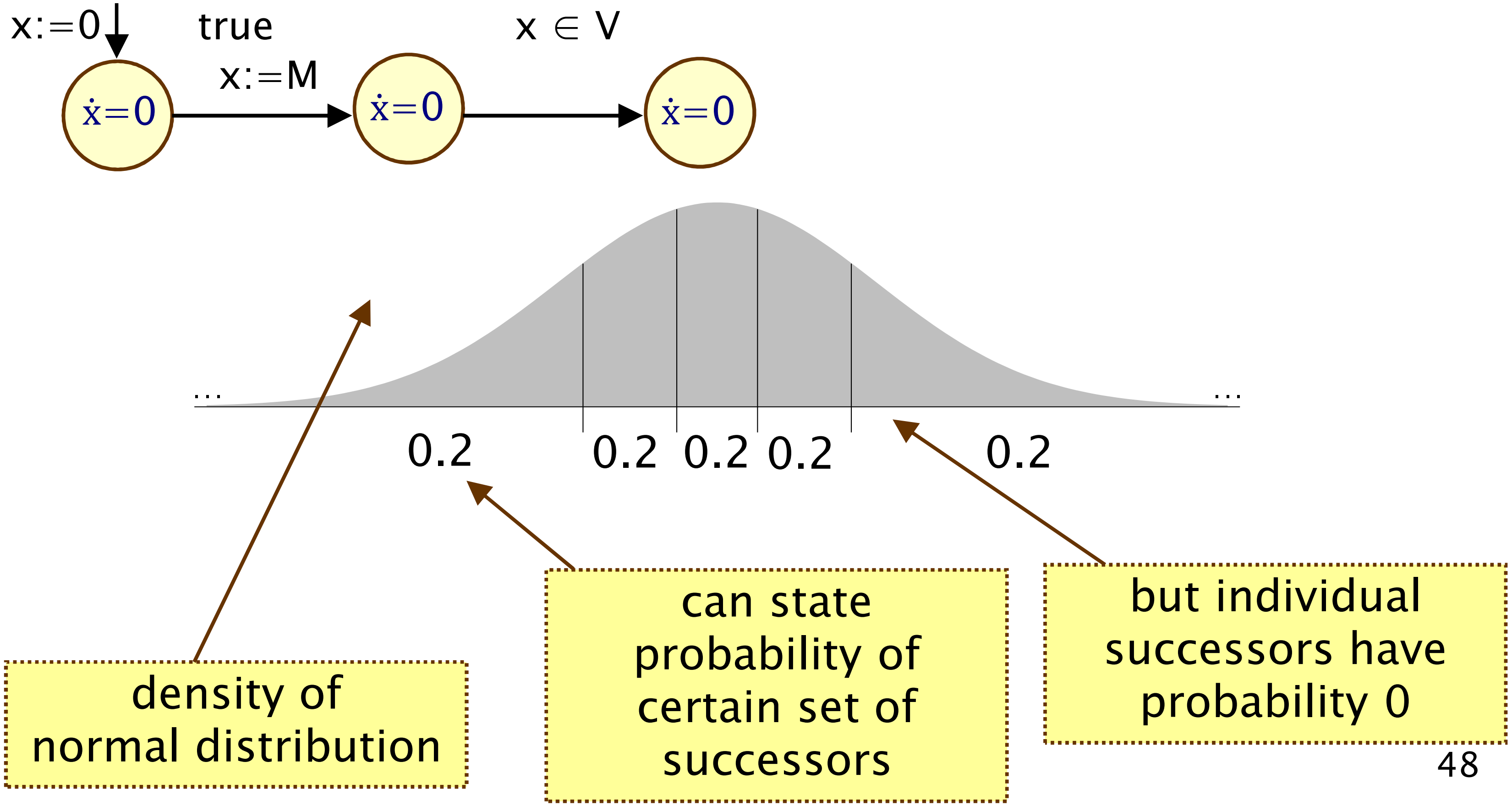


47

Continuous distributions

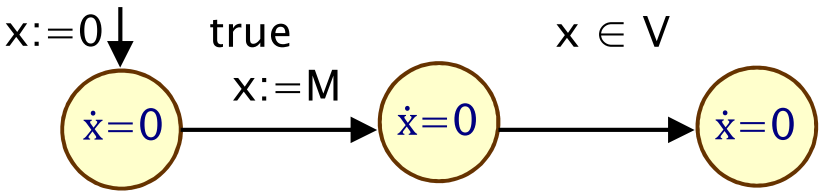
- Often continuous probability distributions of interest
  - Measurements (normal distribution)
  - random delays (exponential distribution)

[FHH+11]



Well-definedness

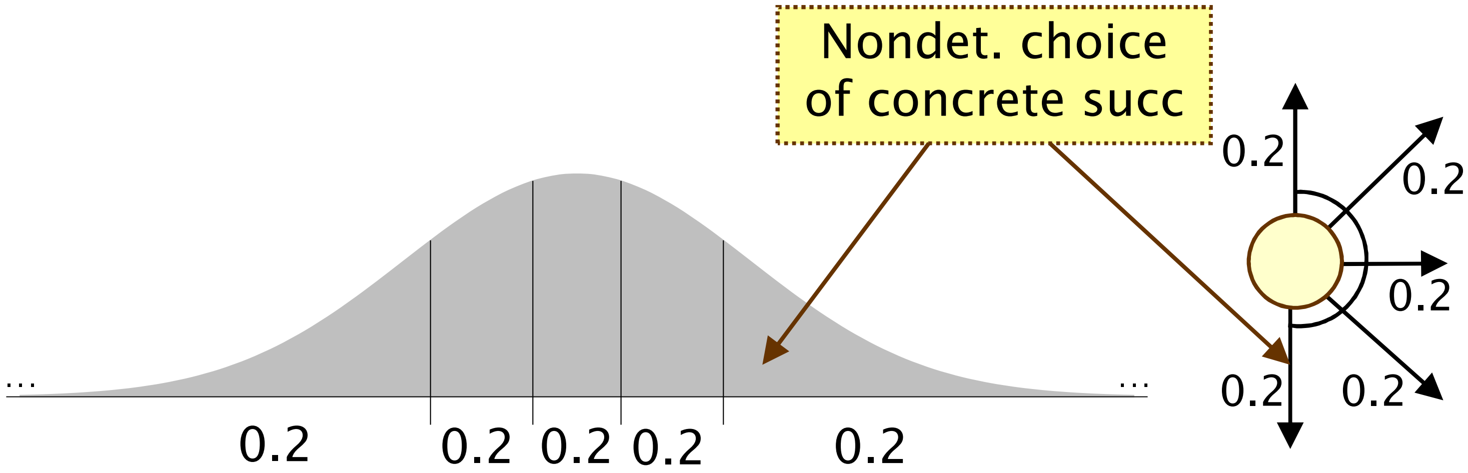
- Semantics: nondeterministic Markov process (NLMP)
- no PA, because of issues with measurability
- restrictions on transitions necessary



- 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

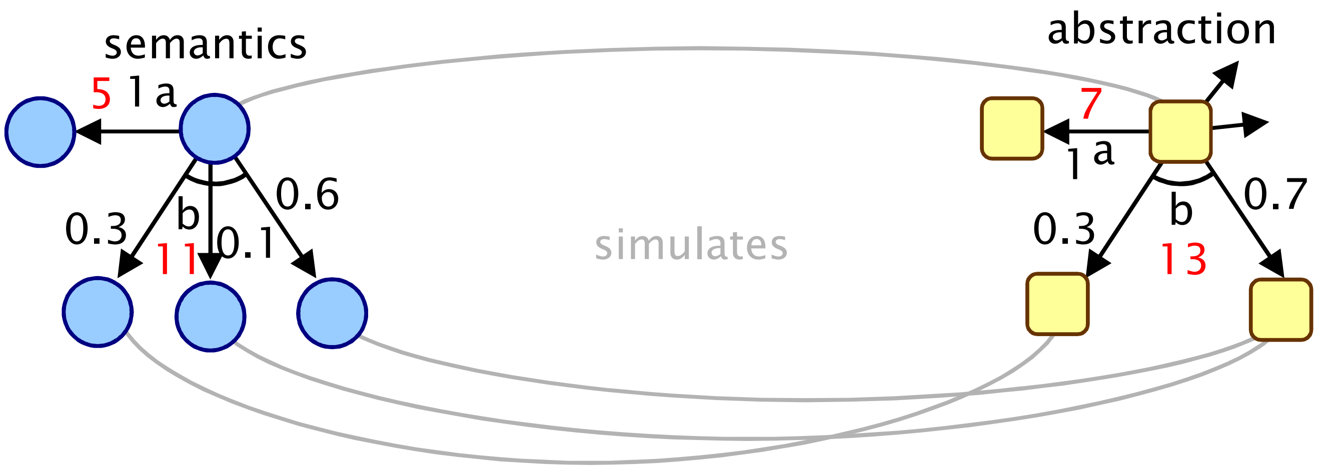
Solution methods

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



Rewards

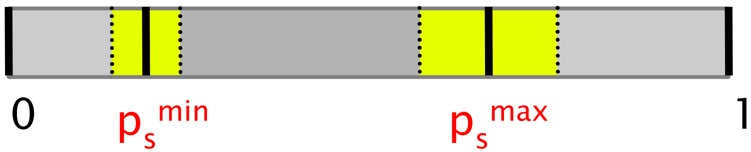
- So far, considered only reachability [HH13]
- Extension to reward-based properties possible
- Extend simulation relation to take reward into account
  - basically, reward in abstraction higher than in semantics
- Extend abstraction by reward structure
- Can analyse similar properties as for basic MDPs
  - cumulative, long-run, etc.



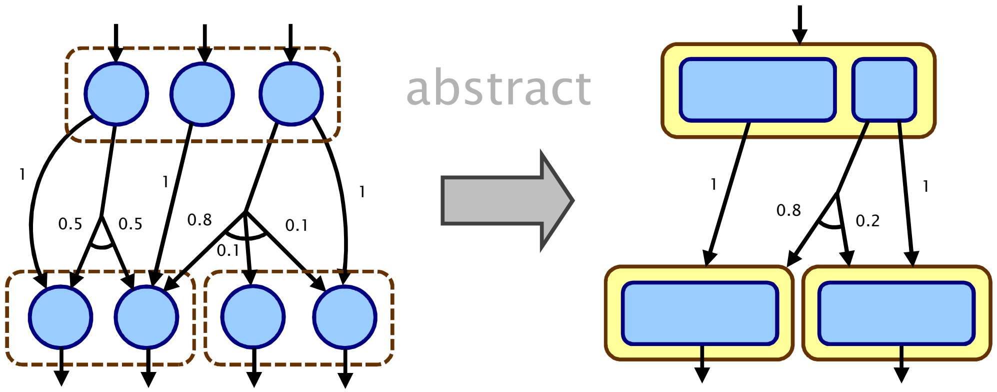
Game-based Abstraction

[HNP+11]

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



- Using similar methods as in the PTA case



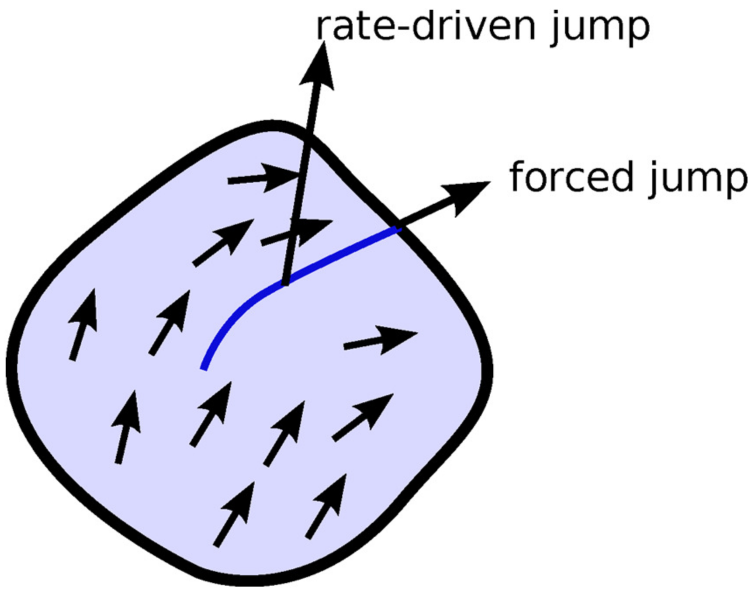
Other notions of PHAs

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

	SHA	STA	PHA	PPTA	PTA	PA	SHS	DTSHS	NLMP
discrete stochastics	✔	✔	✔	✔	✔	✔	✔	✔	✔
continuous stochastics	✔	✔	✗	✗	✗	✗	✔	✔	✔
discrete dynamics	✔	✔	✔	✔	✔	✔	✔	✔	✔
real time	✔	✔	✔	✔	✔	✗	✔	✗	✗
differential inclusions	✔	✗	✔	✗	✗	✗	✗	✗	✗
stochastic differential eqs.	✗	✗	✗	✗	✗	✗	✔	✗	✗
discrete nondeterminism	✔	✔	✔	✔	✔	✔	✗	✗	✔
continuous nondeterminism	✔	✔	✔	✔	✔	✗	✗	✗	✔

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

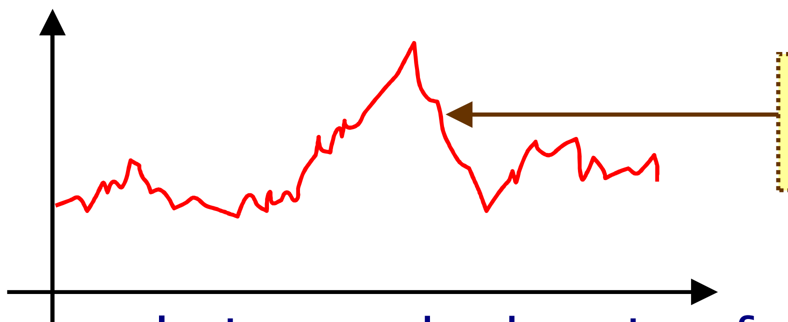
[HLS00]

- No nondeterminism
- But stochastic differential equations within modes
  - $dX(t) = f(Q(\tau_n),X(t))\,dt + g(Q(\tau_n),X(t))\,dB_t$

Standard integral

Stochastic integral

Brownian motion



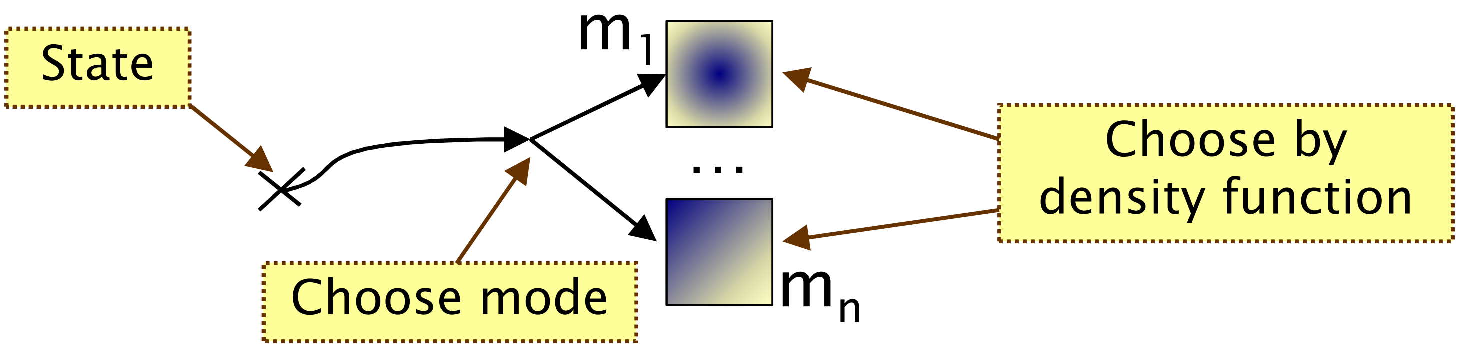
Stochastic process

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

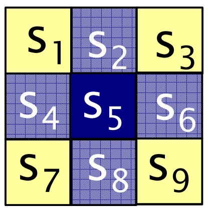
Discrete-time SHS by Abate et al

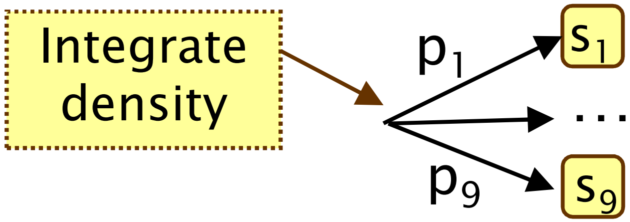
[APLS08]

- 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





PHA model checking – Summary

- 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