Quantitative verification techniques for probabilistic software

Marta Kwiatkowska

Oxford University Computing Laboratory

Summer School on Model Checking, Beijing, October 2010

Course overview

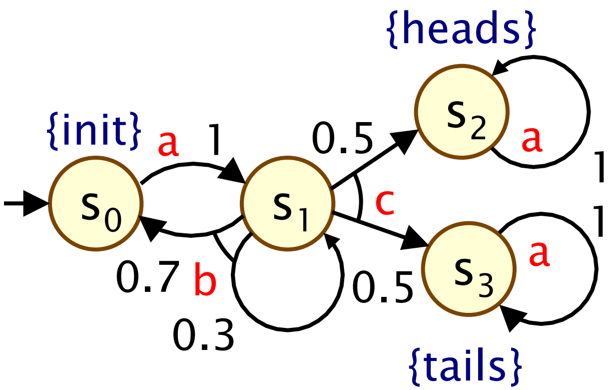
- 3 sessions (Mon/Tue/Thur): 6 $\times$ 50 minute lectures
  - 1: Markov decision processes (MDPs)
  - 2: Probabilistic LTL model checking
  - 3: Compositional probabilistic verification
  - 4: Abstraction, refinement and probabilistic software
  - 5: Probabilistic timed automata (PTAs)
  - 6: Software with time and probabilities
- For additional background material
  - and an accompanying list of references
  - see: http://www.prismmodelchecker.org/lectures/

Part 5

Probabilistic timed automata

Recap: MDPs

- Markov decision processes (MDPs)
  - mix 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 5)

- Time, clocks and zones
- Probabilistic timed automata (PTAs)
  - definition, examples, semantics, time divergence
- PTCTL: A temporal logic for for PTAs
  - syntax, examples, semantics
- Model checking for PTAs
  - the region graph
  - digital clocks

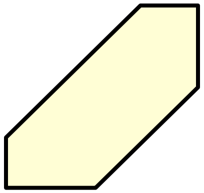
Time, clocks and clock valuations

- 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):

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



  - 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. $x>5 \equiv \neg(x\leq5)$…

- 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

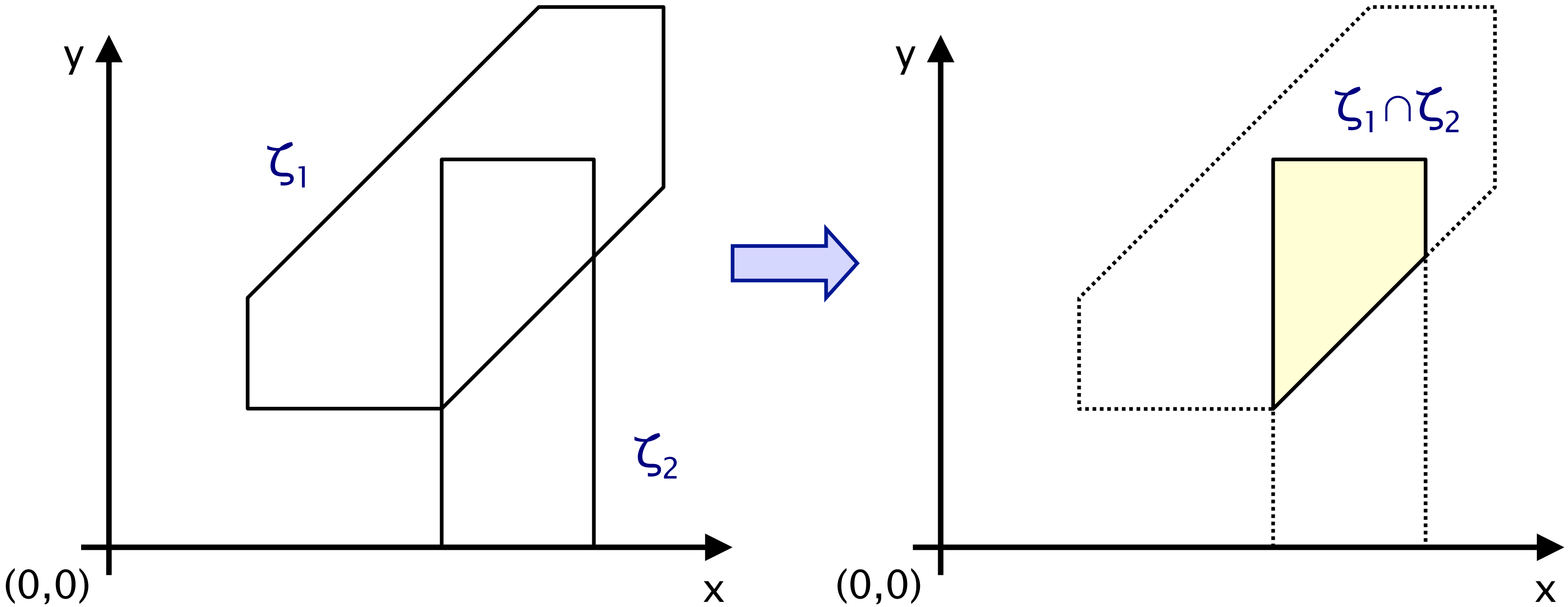
c-equivalence and c-closure

- 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

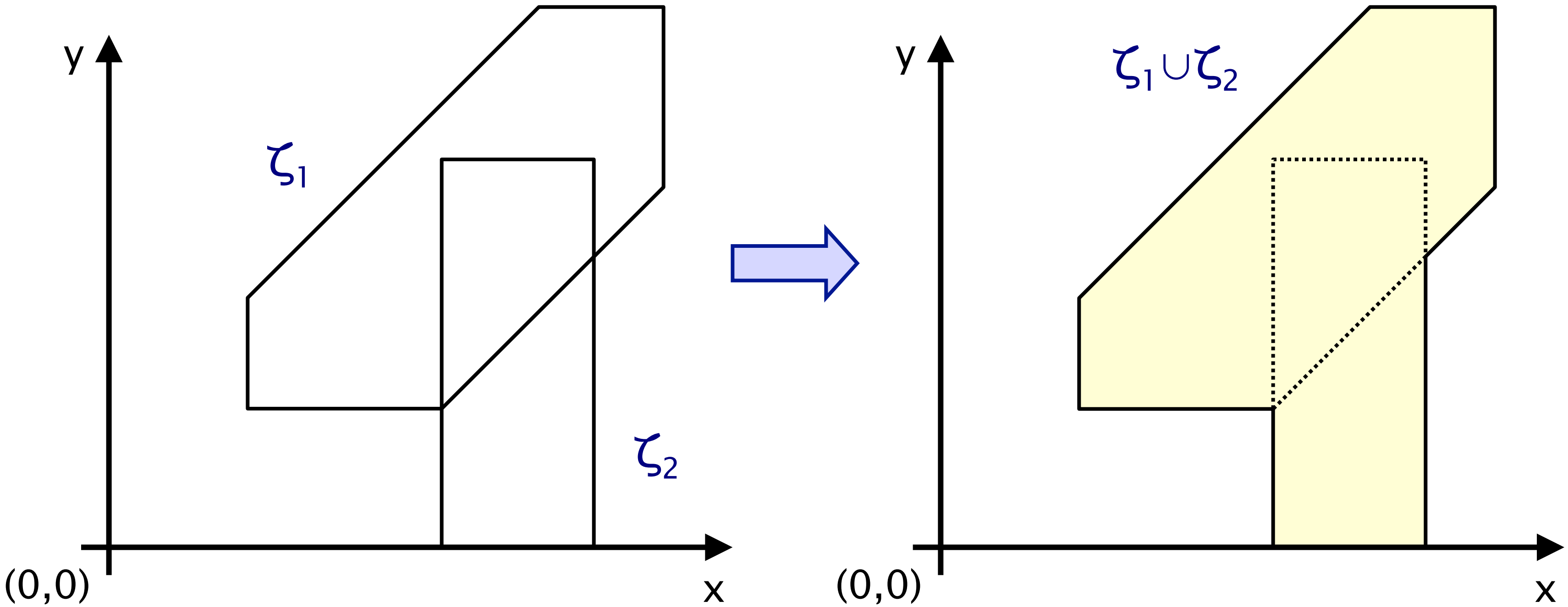
Operations on zones – Set theoretic

- Intersection of two zones: $\zeta_1 \cap \zeta_2$



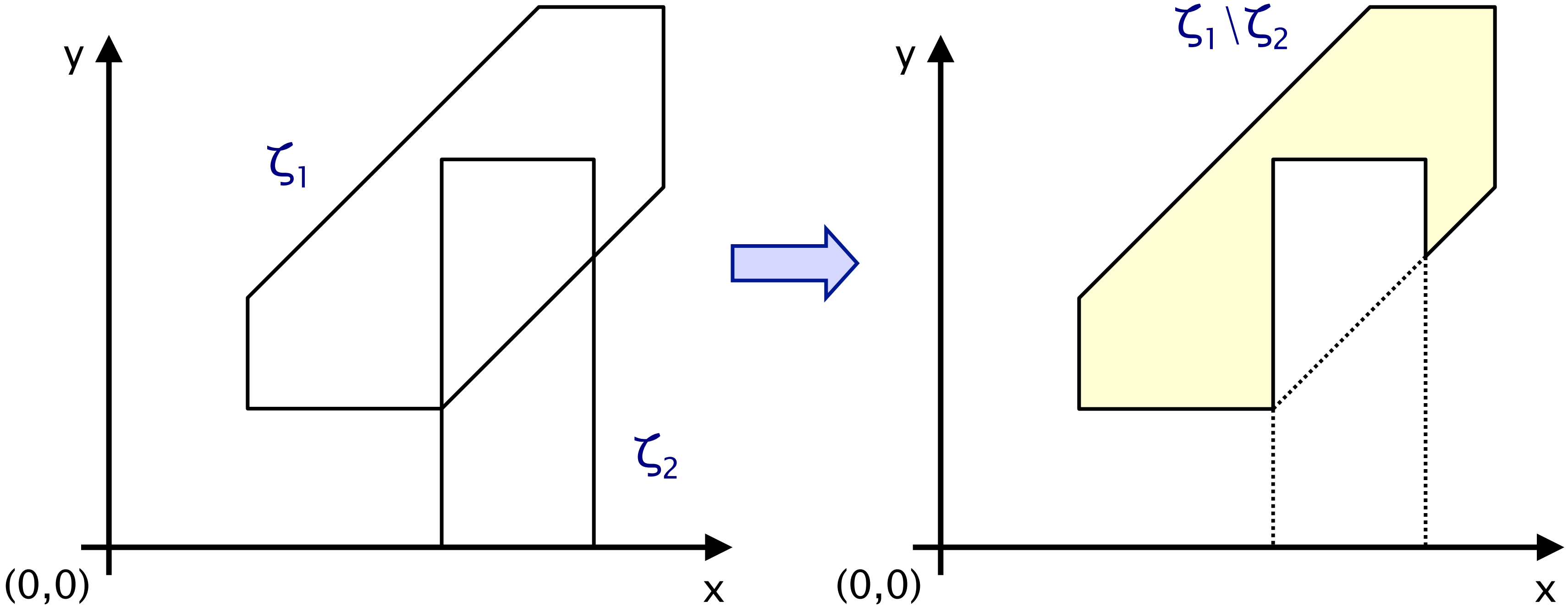
Operations on zones – Set theoretic

- Union of two zones: $\zeta_1 \cup \zeta_2$



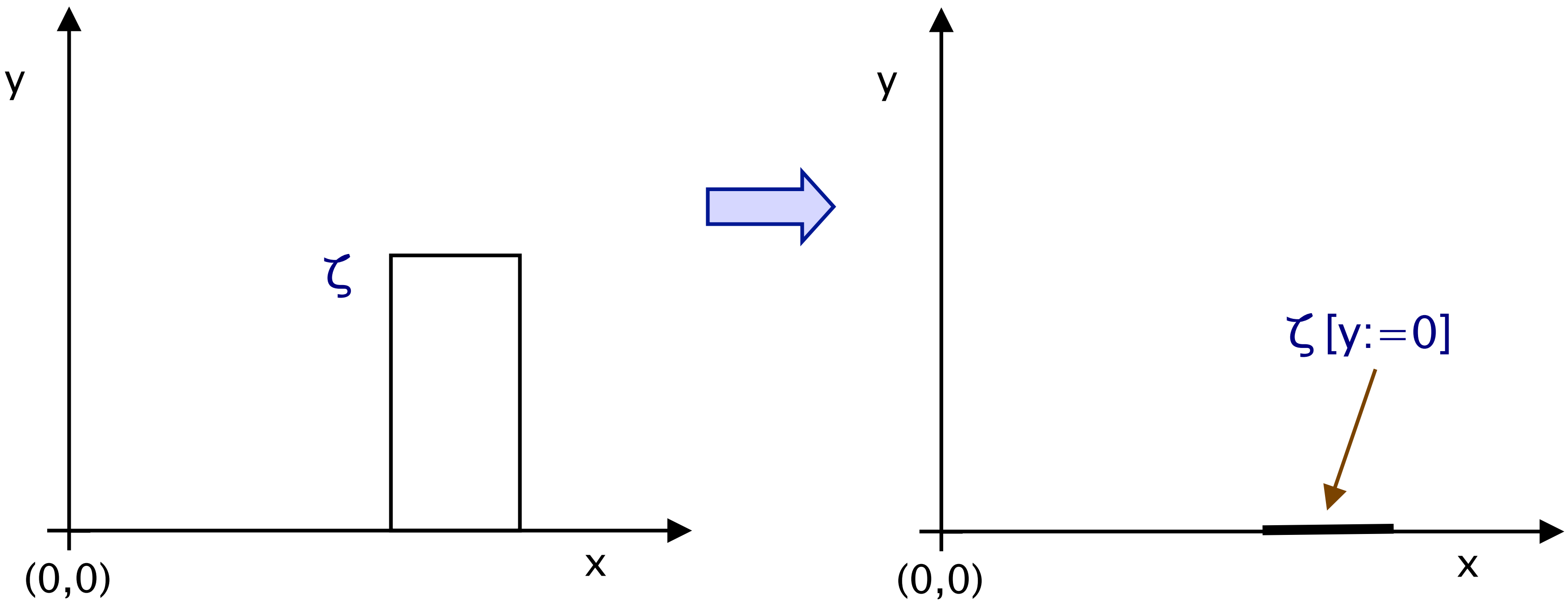
Operations on zones – Set theoretic

- Difference of two zones: $\zeta_1 \setminus \zeta_2$



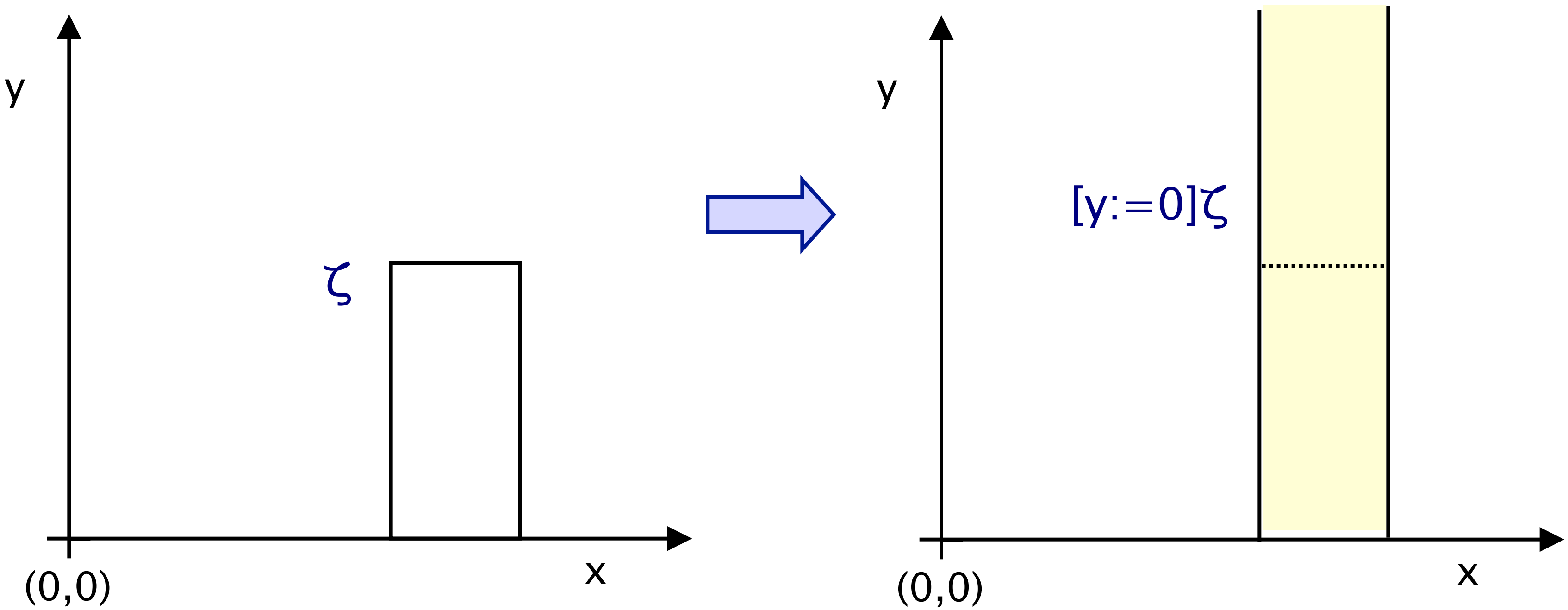
Operations on zones – Clock resets

- $\zeta[Y{:=}0] = \{\, v[Y{:=}0] \mid v \triangleright \zeta \,\}$
  - clock valuations obtained from ζ by resetting the clocks in Y



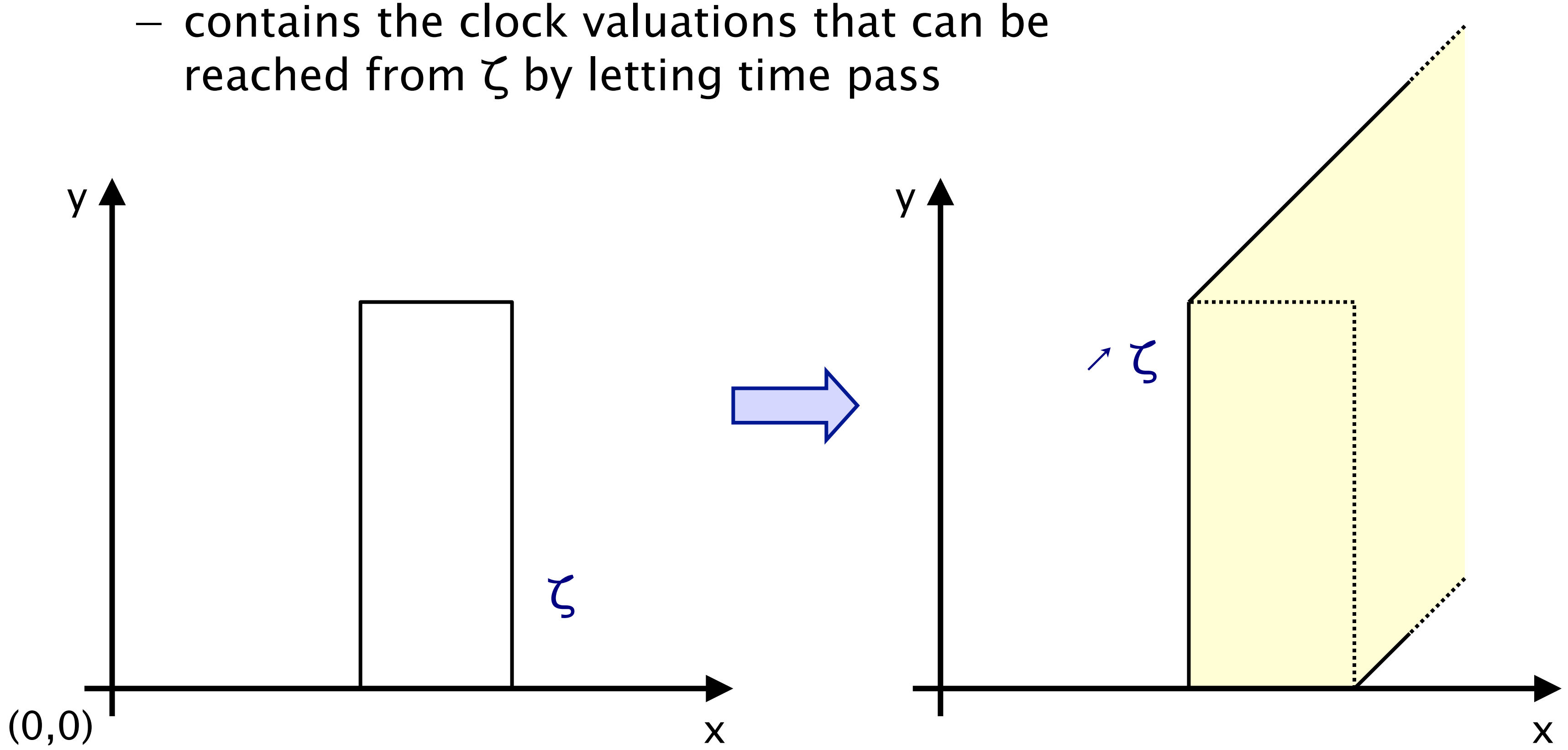
Operations on zones – Clock resets

- $[Y{:=}0]\zeta = \{\, v \mid v[Y{:=}0] \triangleright \zeta \,\}$
  - clock valuations which are in ζ if the clocks in Y are reset



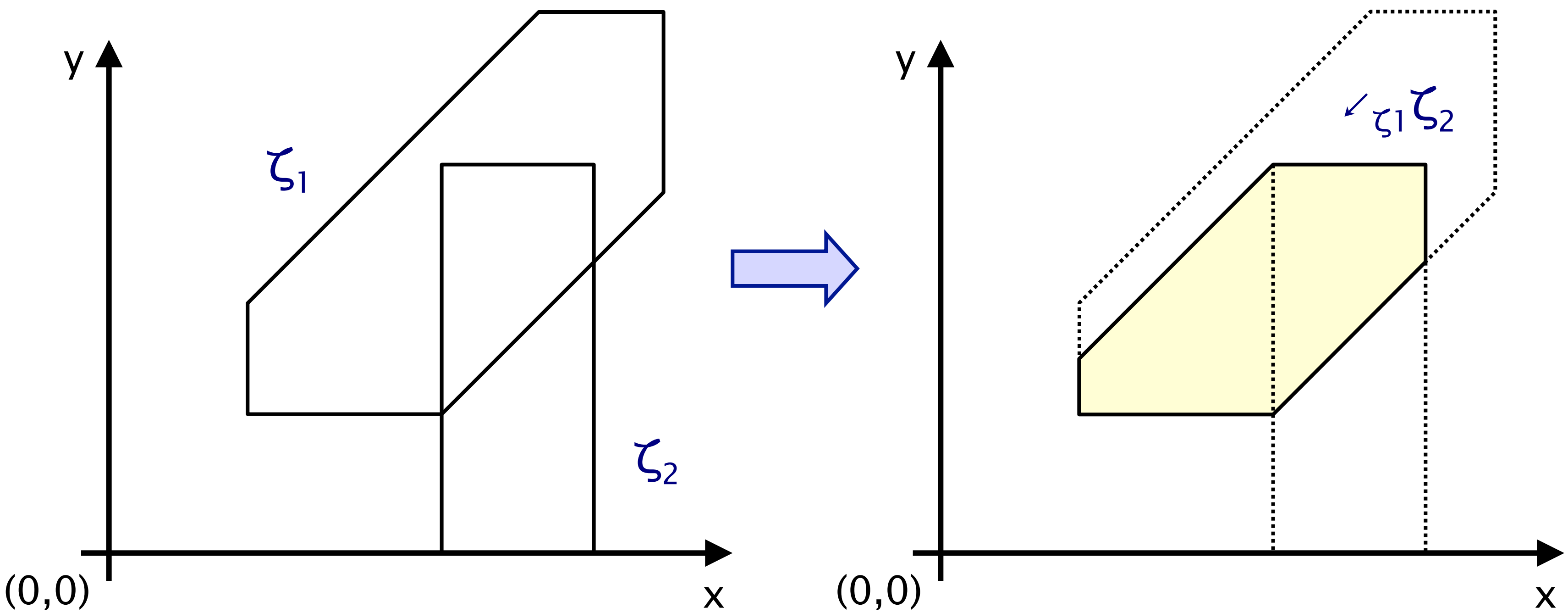
Operations on zones: Projections

- Forwards diagonal projection
- $\nearrow\zeta = \{ v \mid \exists t \geq 0 \,.\, (v-t) \triangleright \zeta \}$
  - contains the clock valuations that can be reached from ζ by letting time pass



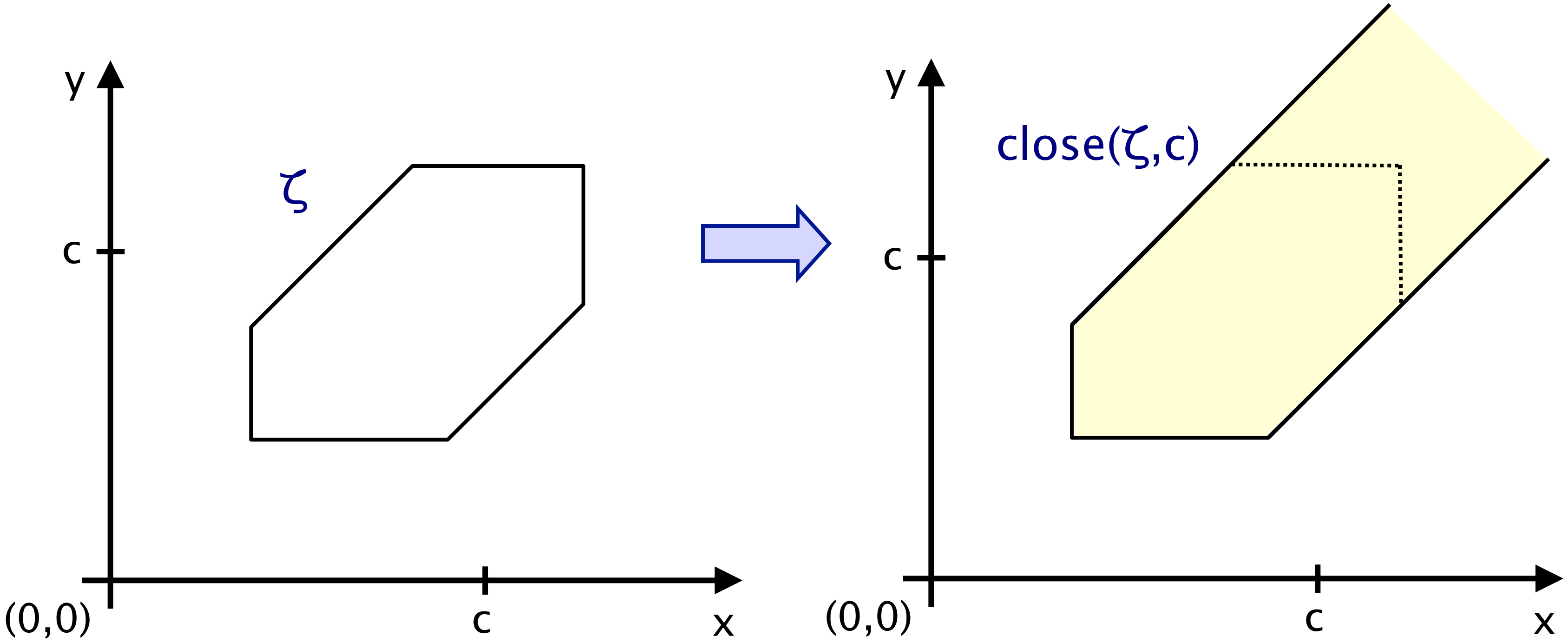
Operations on zones: Projections

- Backwards diagonal projection
- $\swarrow_{\zeta'} \zeta = \{ v \mid \exists t \geq 0 \,.\, ((v+t) \triangleright \zeta \wedge \forall t' < t \,.\, ((v+t') \triangleright \zeta')) \}$
  - contains the clock valuations that, by letting time pass, reach a clock valuation in ζ and remain in ζ' until ζ is reached



Operations on zones: c-closure

- c-closure: close(ζ,c)
  - ignores all constraints which are greater than c



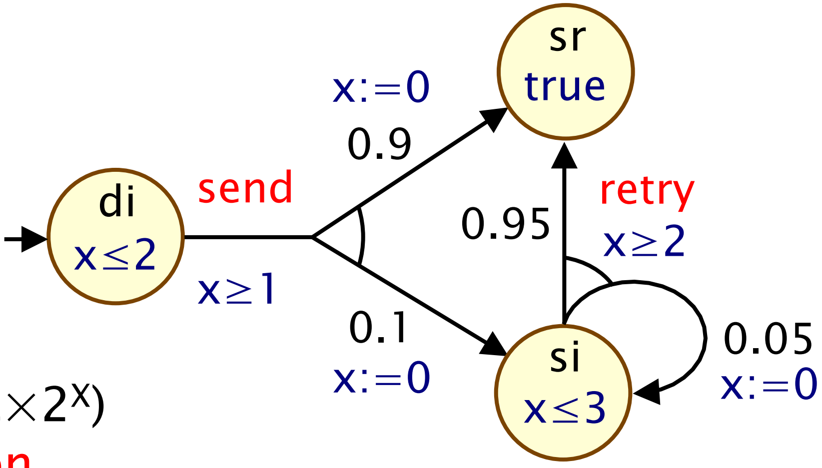
Overview (Part 5)

- Time, clocks and zones
- Probabilistic timed automata (PTAs)
  - definition, examples, semantics, time divergence
- PTCTL: A temporal logic for for PTAs
  - syntax, examples, semantics
- Model checking for PTAs
  - the region graph
  - digital clocks

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

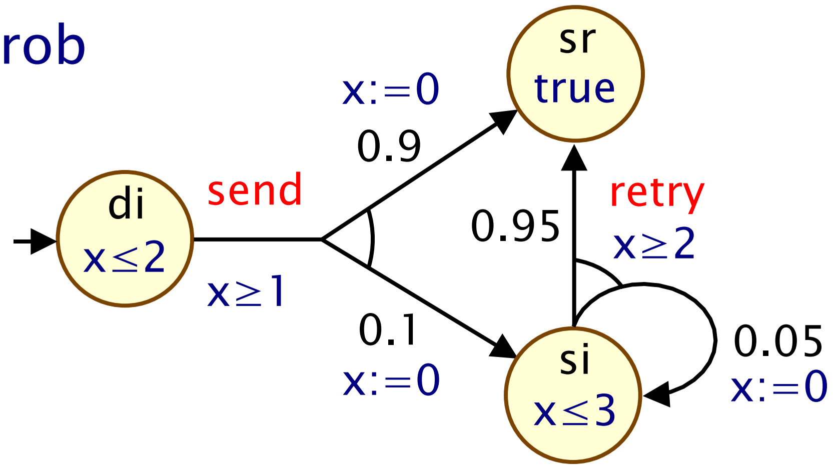
- Syntax: A PTA is a tuple (Loc, l_{init}, Act, X, inv, prob, L)
  - Loc is a finite set of locations
  - $l_{init} \in$ Loc is the initial location
  - Act is a finite set of actions
  - X is a finite set of clocks
  - inv : Loc → Zones(X) is the invariant condition
  - prob ⊆ Loc×Zones(X)×Dist(Loc×2^X)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



Probabilistic edge relation

- Probabilistic edge relation
  - prob ⊆ Loc×Zones(X)×Act×Dist(Loc×2^X)

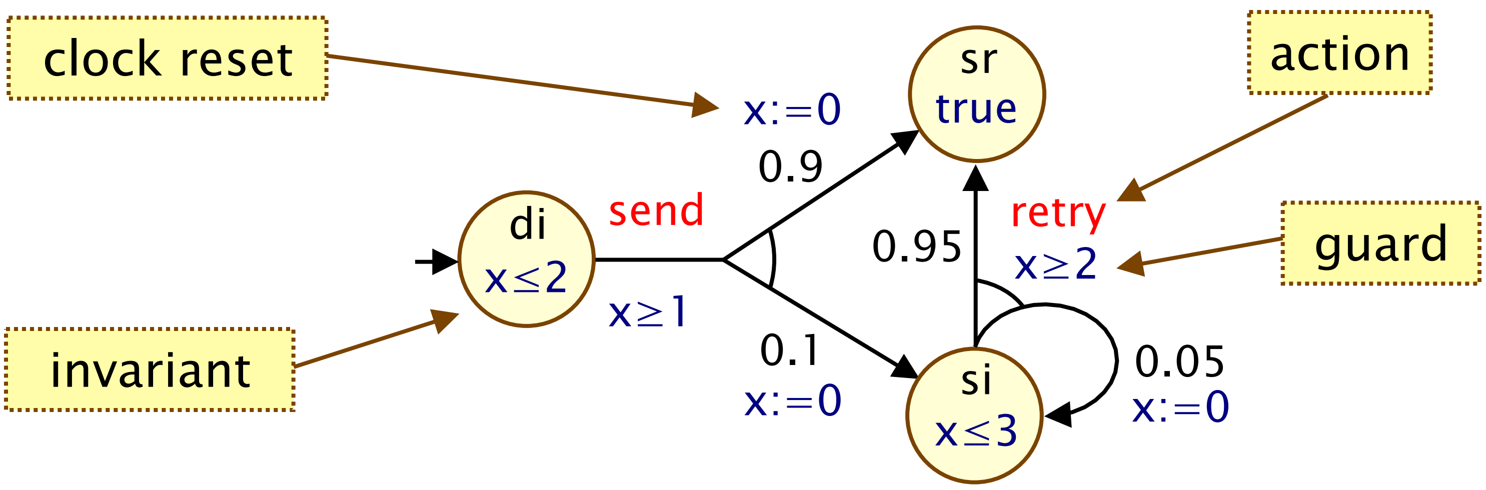
- Probabilistic edge (l,g,a,p) ∈ prob
  - l is the source location
  - g is the guard
  - a is the action
  - p target distribution



- 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 (to zero)

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



PTAs – Behaviour

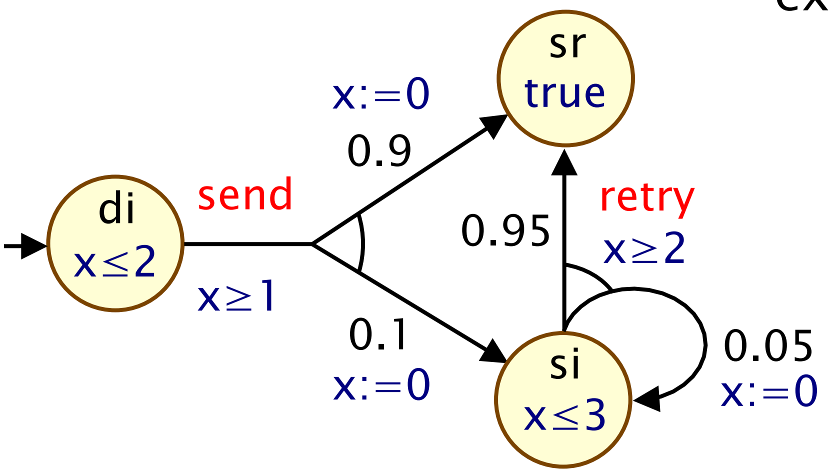
- A state of a PTA is a pair (l,v) $\in \mathrm{Loc} \times \mathbb{R}^{X}$ such that v $\triangleright$ inv(l)

- 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

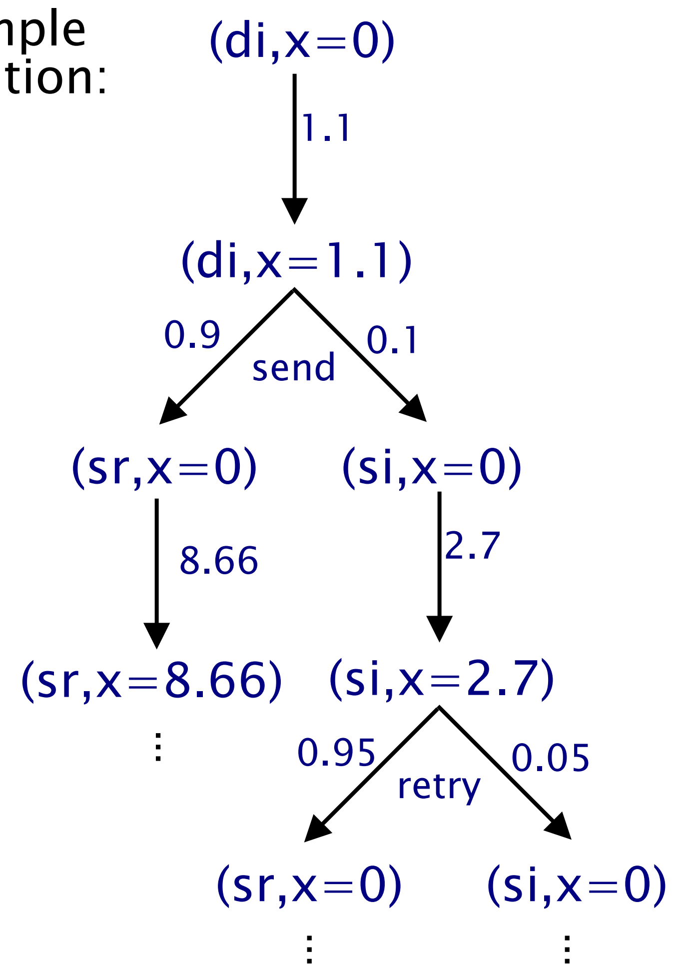
- For any state (l,v), there is nondeterministic choice between making a discrete transition and letting time pass
  - discrete transition (l,g,a,p) enabled if v $\triangleright$ g and probability of moving to location l' and resetting the clocks Y equals p(l',Y)
  - time transition available only if 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 $M_P = (S_P, s_{init}, \alpha_P, \delta_P, L_P)$ with:

- States: $S_P = \{ (l,v) \in Loc \times \mathbb{R}^X$ such that $v \rhd inv(l) \}$
- Initial state: $s_{init} = (l_{init}, \underline{0})$

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

- Actions: $\alpha_P = Act \cup \mathbb{R}$
- $\delta_P \subseteq S_P \times \alpha_P \times Dist(S_P)$ such that $(s, a, \mu) \in \delta_P$ iff:
  - (time transition) $a \in \mathbb{R}$, $\mu(l,v+t)=1$ and $v+t' \rhd inv(l)$ for all $t' \leq t$
  - (discrete transition) $a \in Act$ and there exists $(l,g,a,p) \in prob$

    such that $v \rhd g$ 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

Time divergence

- We restrict our attention to time divergent behaviour
  - a common restriction imposed in real-time systems
  - unrealisable behaviour (i.e. corresponding to time not advancing beyond a time bound) is disregarded
  - also called non-zeno behaviour

- For a path $\omega = s_0(a_0,\mu_0)s_1(a_1,\mu_1)s_2(a_2,\mu_2)\ldots$ in the MDP M_P
  - $D_\omega(n)$ denotes the duration up to state s_n
  - i.e. $D_\omega(n) = \Sigma \,\{|\; a_i \mid 0 \leq i < n \wedge a_i \in \mathbb{R} \;|\}$

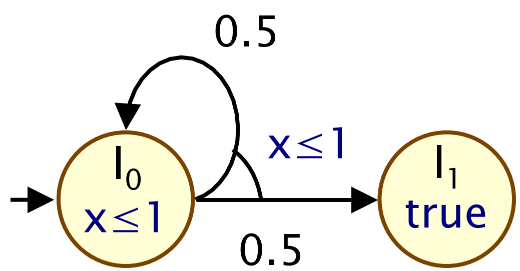
- A path ω is time divergent if, for any $t \in \mathbb{R}_{\geq 0}$:
  - there exists $j \in \mathbb{N}$ such that $D_\omega(j) > t$

- Example of non-divergent path:
  - $s_0(1,\mu_0)s_0(0.5,\mu_0)s_0(0.25,\mu_0)s_0(0.125,\mu_0)s_0\ldots$

Time divergence

- An adversary of M_P is divergent if, for each state $s \in S_P$:
  - the probability of divergent paths under A is 1
  - i.e $Pr^A_s\{\ \omega \in Path^A(s) \mid \omega \text{ is divergent }\} = 1$

- Motivation for probabilistic definition of divergence:



  - in this PTA, any adversary has one non-divergent path:
    - takes the loop in l_0 infinitely often, without 1 time unit passing
  - but the probability of such behaviour is 0
  - a stronger notion of divergence would mean no divergent adversaries exist for this PTA

Overview (Part 5)

- Time, clocks and zones
- Probabilistic timed automata (PTAs)
  - definition, examples, semantics, time divergence
- PTCTL: A temporal logic for for PTAs
  - syntax, examples, semantics
- Model checking for PTAs
  - the region graph
  - digital clocks

PTCTL – Syntax

- PTCTL: Probabilistic timed computation tree logic
  - derived from PCTL [BdA95] and TCTL [AD94]

- Syntax:

ϕ U ϕ is true with probability ~p

  - $\phi ::= \mathrm{true} \mid a \mid \zeta \mid z.\,\phi \mid \phi \wedge \phi \mid \neg\phi \mid P_{\sim p}\,[\,\phi \;\mathrm{U}\; \phi\,]$

"zone over X∪Z"

"freeze quantifier"

- where:
  - where Z is a set of formula clocks, $\zeta \in \mathrm{Zones}(X \cup Z)$, $z \in Z$,
  - a is an atomic proposition, $p \in [0,1]$ and $\sim \in \{<,>,\leq,\geq\}$

PTCTL – Examples

- $z\,.\,P_{>0.99}\,[\,\mathrm{packet2unsent}\ \mathrm{U}\ \mathrm{packet1delivered} \wedge (z<5)\,]$
  – "with probability greater than 0.99, the system delivers packet 1 within 5 time units and does not try to send packet 2 in the meantime"

- $z\,.\,P_{>0.95}[\,(x\leq 3)\ \mathrm{U}\ (z=8)\,]$
  – "with probability at least 0.95, the system clock x does not exceed 3 before 8 time units elapse"

- $z\,.\,P_{\leq 0.1}[\,\mathrm{G}\ (\mathrm{failure} \vee (z\leq 60))\,]$
  – "the system fails after the first 60 time units have elapsed with probability at most 0.01"

PTCTL – Semantics

- Let $(l,v) \in S_p$ and $\mathcal{E} \in \mathbb{R}^Z$ be a formula clock valuation

combined clock valuation of v and $\mathcal{E}$ satisfies ζ

after resetting z, ϕ is satisfied

- $(l,v),\mathcal{E} \models a \quad \Leftrightarrow \quad a \in L(l,v)$
- $(l,v),\mathcal{E} \models \zeta \quad \Leftrightarrow \quad v,\mathcal{E} \triangleright \zeta$
- $(l,v),\mathcal{E} \models z.\phi \quad \Leftrightarrow \quad (l,v),\mathcal{E}[z:=0] \models \phi$
- $(l,v),\mathcal{E} \models \phi_1 \wedge \phi_2 \quad \Leftrightarrow \quad (l,v),\mathcal{E} \models \phi_1$ and $(l,v),\mathcal{E} \models \phi_2$
- $(l,v),\mathcal{E} \models \neg\phi \quad \Leftrightarrow \quad (l,v),\mathcal{E} \models \phi$ is false
- $(l,v),\mathcal{E} \models P_{\sim p}[\psi] \quad \Leftrightarrow \quad Pr^A_{(l,v)}\{ \omega \in Path^A(l,v) \mid \omega,\mathcal{E} \models \psi \} \sim p$ for all adversaries $A \in Adv_{M_P}$

the probability of a path satisfying ψ meets $\sim p$ for all divergent adversaries

PTCTL – Semantics of until

- Let ω be a path in M_P and $\mathcal{E}$ be a formula clock valuation
  - $\omega,\mathcal{E} \models \psi$ satisfaction of ψ by ω, assuming $\mathcal{E}$ initially

- $\omega,\mathcal{E} \models \phi_1 \ U \ \phi_2$ if and only if
  there exists $i \in \mathbb{N}$ and $t \in D_\omega(i+1)-D_\omega(i)$ such that
  - $\omega(i)+t,\mathcal{E}+(D_\omega(i)+t) \models \phi_2$
  - $\forall\ t' \leq t\ .\ \omega(i)+t',\mathcal{E}+(D_\omega(i)+t') \models \phi_1 \vee \phi_2$
  - $\forall\ j<i\ .\ \forall\ t' \leq D_\omega(j+1)-D_\omega(j)\ .\ \omega(j)+t',\mathcal{E}+(D_\omega(j)+t') \models \phi_1 \vee \phi_2$

- Condition "$\phi_1 \vee \phi_2$" different from PCTL and CSL
  - usually ϕ_2 becomes true and ϕ_1 is true until this point
  - difference due to the density of the time domain
  - to allow for open intervals use disjunction $\phi_1 \vee \phi_2$
  - for example consider $x \leq 5\ U\ x>5$ and $x<5\ U\ x \geq 5$

Probabilistic reachability in PTAs

- For simplicity, in some cases, we just consider probabilistic reachability, rather than full PTCTL 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 5)

- Time, clocks and zones
- Probabilistic timed automata (PTAs)
  - definition, examples, semantics, time divergence
- PTCTL: A temporal logic for for PTAs
  - syntax, examples, semantics
- Model checking for PTAs
  - the region graph
  - digital clocks

PTA model checking – Summary

- Several different approaches developed
  - basic idea: reduce to the analysis of a finite-state model
  - in most cases, this is a Markov decision process (MDP)

- Region graph construction [KNSS02]
  - shows decidability, but gives exponential complexity
- Digital clocks approach [KNPS06]
  - (slightly) restricted classes of PTAs
  - works well in practice, still some scalability limitations
- Zone-based approaches (next lecture)
  - (preferred approach for non-probabilistic timed automata)
  - forwards reachability [KNSS02]
  - backwards reachability [KNSW07]
  - game-based abstraction refinement [KNP09c]

The region graph

- Region graph construction for PTAs [KNSS02]
  - adapts region graph construction for timed automata [ACD93]
  - partitions PTA states into a finite set of regions
  - based on notion of clock equivalence
  - construction is also dependent on PTCTL formula

- For a PTA P and PTCTL formula ϕ
  - construct a time-abstract, finite-state MDP $R(\phi)$
  - translate PTCTL formula ϕ to PCTL formula ϕ'
  - ϕ is preserved by region quivalence
  - i.e. ϕ holds in a state of M_P if and only if ϕ' holds in the corresponding state of $R(\phi)$
  - model check $R(\phi)$ using standard methods for MDPs

The region graph – Clock equivalence

- Regions are sets of clock equivalent clock valuations

- Some notation:
  - let c be largest constant appearing in PTA or PTCTL formula
  - let $\lfloor t \rfloor$ denotes the integral part of t
  - t and t' agree on their integral parts if and only if

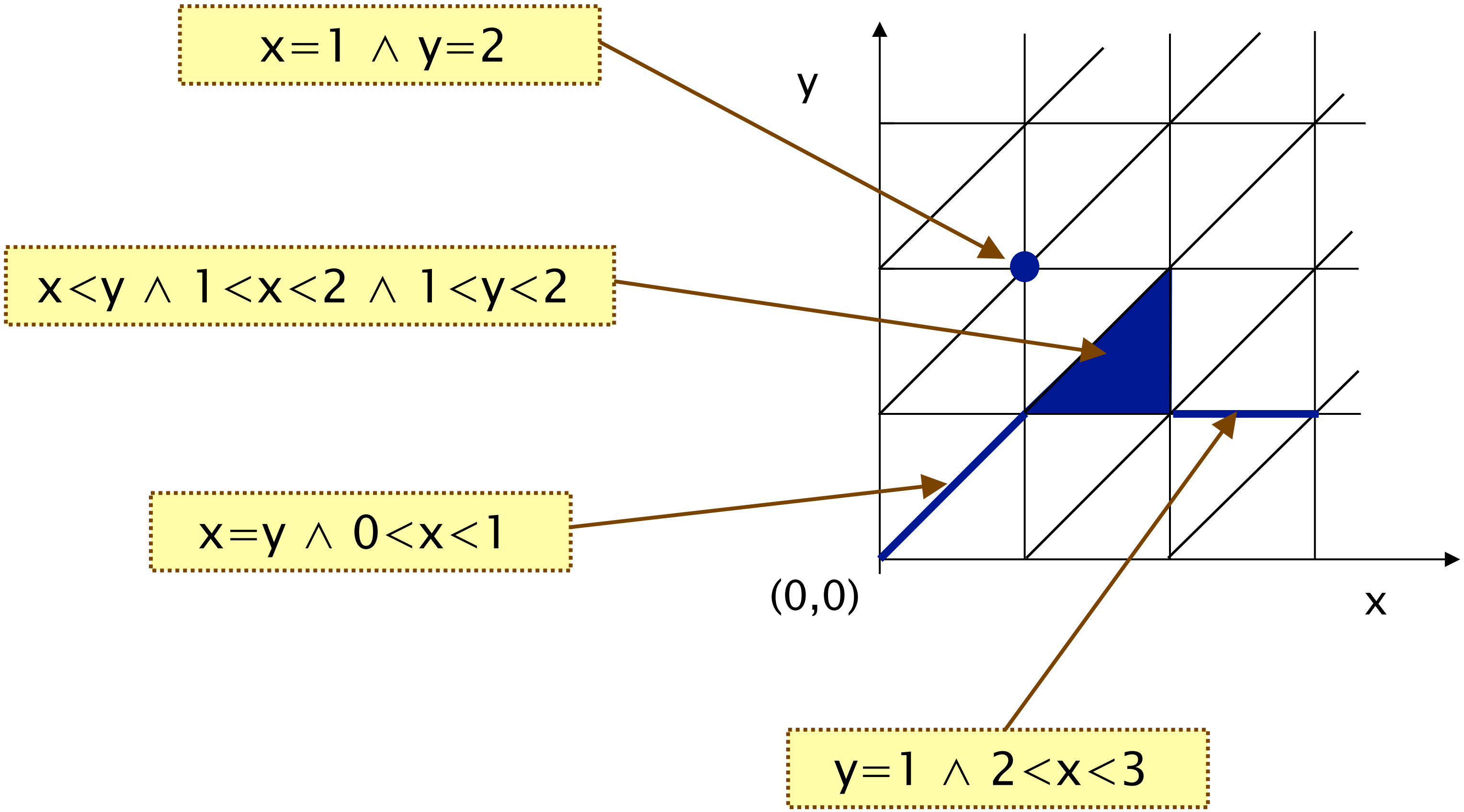
    (1) $\lfloor t \rfloor = \lfloor t' \rfloor$

    (2) t and t' are both integers or neither is an integer

- The clock valuations v and v' are clock equivalent ($v \cong v'$) if:
  - for all clocks $x \in X$, either:
    - v(x) and v'(x) agree on their integral parts
    - $v(x)>c$ and $v'(x)>c$
  - for all clock pairs $x,y \in X$, either:
    - $v(x) - v(x')$ and $v'(x) - v'(x')$ agree on their integral parts
    - $v(x) - v(x') > c$ and $v'(x) - v'(x') > c$

Region graph – Clock equivalence

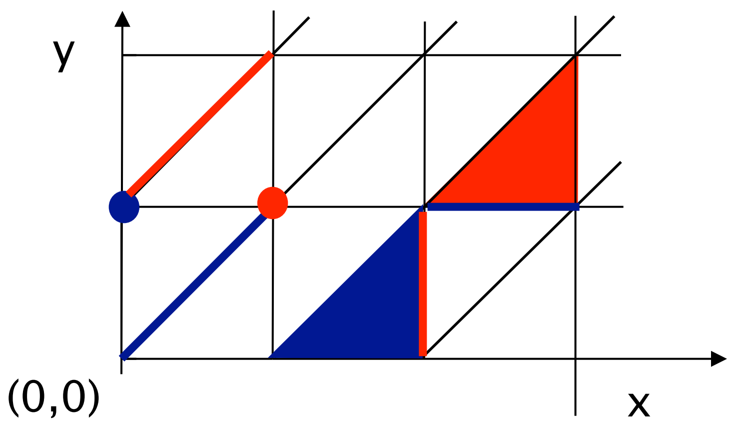
- Example regions (for 2 clocks x and y)



Region graph – Clock equivalence

- Fundamental result: if $v \cong v'$, then $v \triangleright \zeta \Leftrightarrow v' \triangleright \zeta$
  - it follows that $r \triangleright \zeta$ is well defined for a region r

- r' is the successor region of r, written succ(r) = r', if
  - for each $v \in r$, there exists $t>0$ such that $v+t \in r'$ and $v+t' \in r \cup r'$ for all $t' < t$



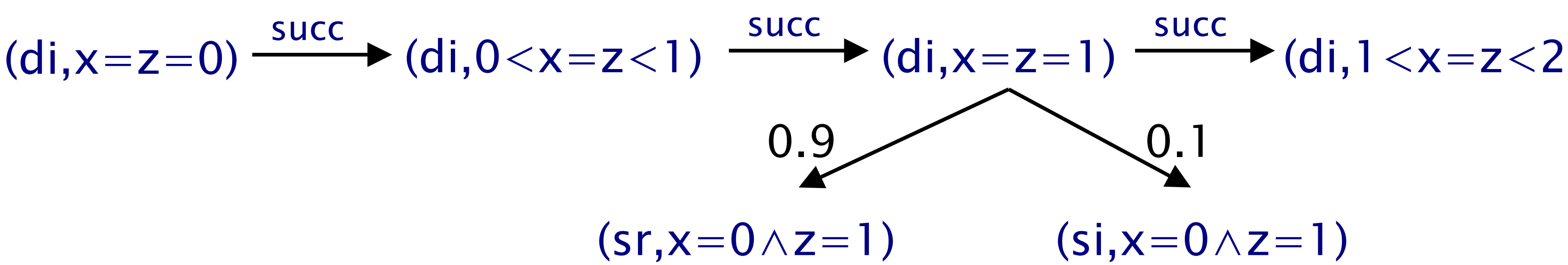
The region graph

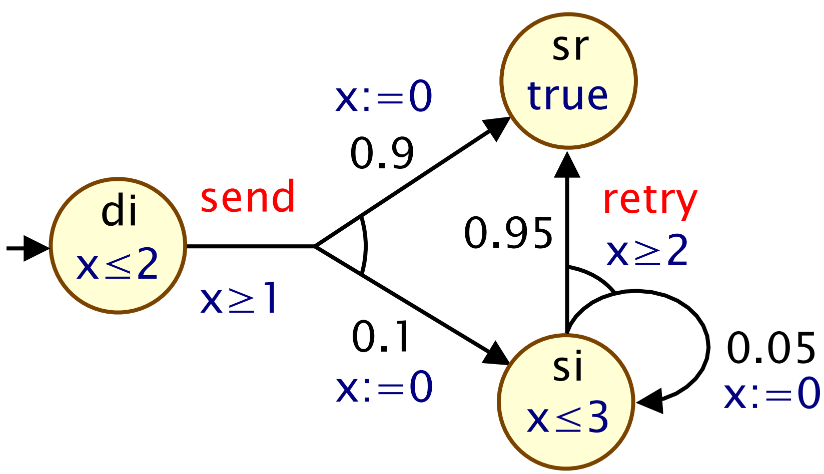
- The region graph MDP is $M_R = (S_R, s_{init}, \alpha_R, \delta_R, L_R)$ where…
  - the set of states S_R comprises pairs (l,r) such that l is a location and r is a region over $X \cup Z$
  - the initial state s_{init} is $(l_{init}, \underline{0})$
  - the set of actions α_R is $\{succ\} \cup Act$
    - succ is a unique action denoting passage of time
  - the probabilistic transition function δ_R is defined as:
  - $((l,r),succ,\mu) \in \delta_R(l,r)$ iff $\mu(l,succ(r))=1$
  - $((l,r),a,\mu) \in \delta_R(l,r)$ iff $\exists$ (l,g,a,p) $\in$ prob such that

    $r \triangleright g$ and, for any $(l',r') \in S_R$: $\mu(l',r') = \sum_{Y \subseteq X \wedge r[Y:=0]=r'} p(l',Y)$
  - the labelling is given by: $L_R(l,r) = L(l)$

Region graph – Example

- PTCTL formula: $z.P_{\sim p}$ [true U (sr<4)]





Region graph construction

- Region graph
  - useful for establishing decidability of model checking
  - or proving complexity results for model checking algorithms

- But…
  - the number of regions is exponential in the number of clocks and the size of largest constant
  - so model checking based on this is extremely expensive
  - and so not implemented (even for timed automata)

- Improved approaches based on:
  - digital clocks
  - zones (unions of regions)

Overview (Part 5)

- Time, clocks and zones
- Probabilistic timed automata (PTAs)
  - definition, examples, semantics, time divergence
- PTCTL: A temporal logic for for PTAs
  - syntax, examples, semantics
- Model checking for PTAs
  - the region graph
  - digital clocks

Digital clocks

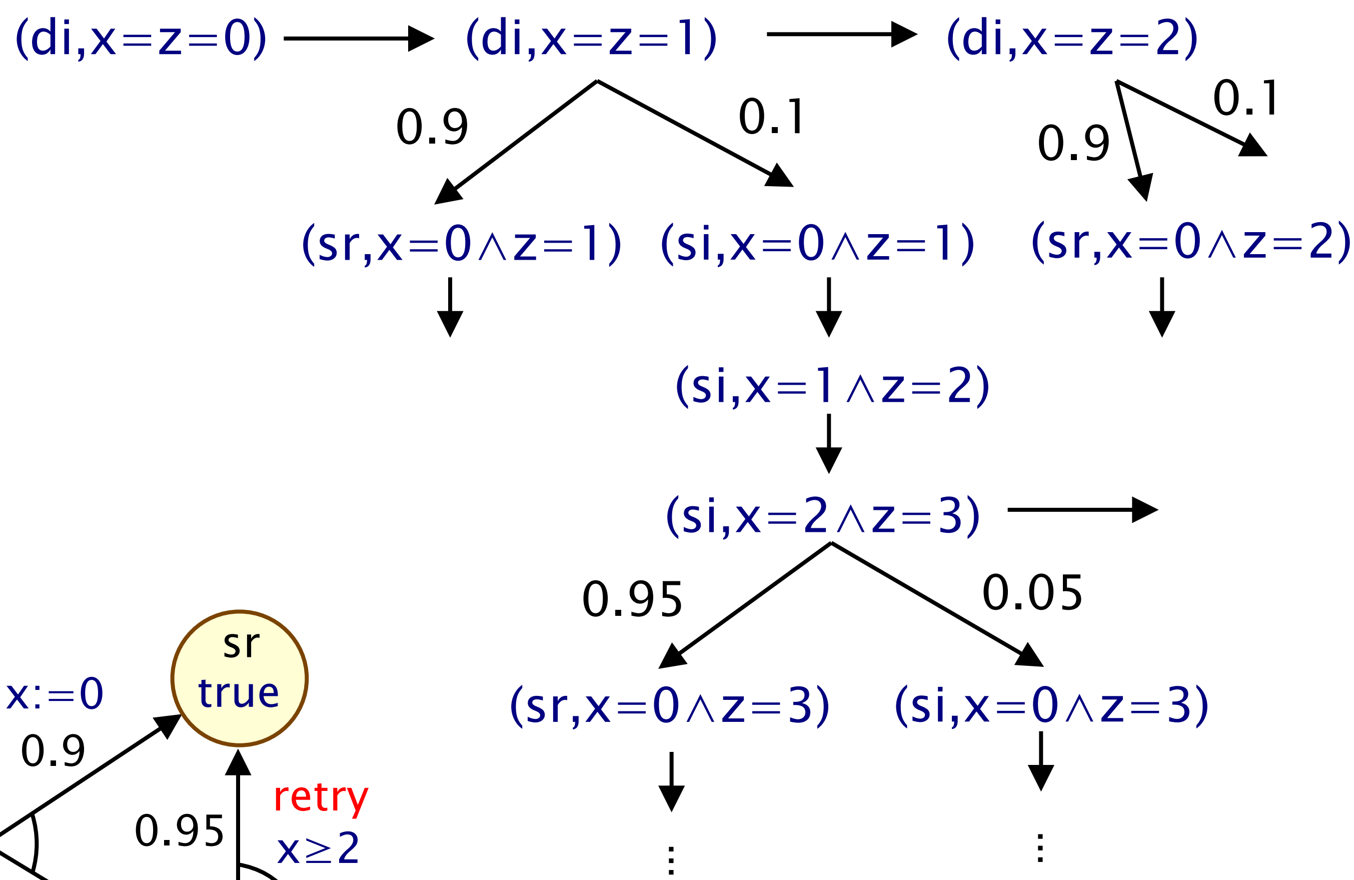
- Simple idea: Clocks can only take integer (digital) values
  - i.e. time domain is $\mathbb{N}$ as opposed to $\mathbb{R}$
  - based on notion of ϵ-digitisation [HMP92]

- Only applies to arestricted class of PTAs; zones must be:
  - closed – no strict inequalities (e.g. $x>5$)

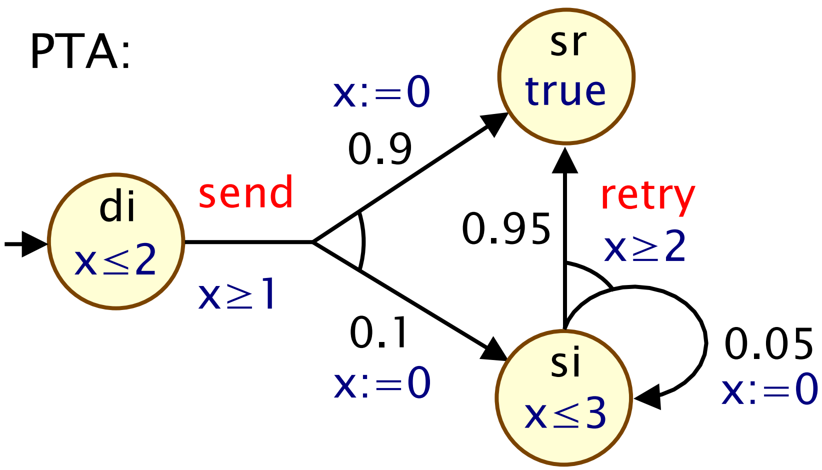
- Digital clocks semantics yields a finite-state MDP
  - state space is a subset of $Loc \times \mathbb{N}^X$, rather than $Loc \times \mathbb{R}^X$
  - clocks bounded by c_{max} (max constant in PTA and formula)
  - then use standard techniques for finite -state MDPs

Example – Digital clocks

MDP:
(digital
clocks)



PTA:



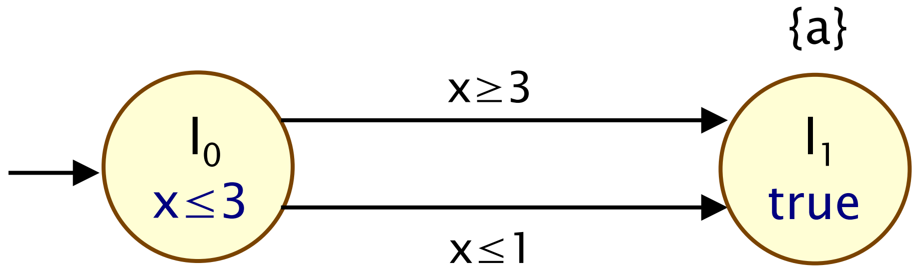
Digital clocks

- Digital clocks approach preserves:
  - minimum/maximum reachability probabilities
  - a subset of PTCTL properties
  - (no nesting, only closed zones in formulae)
  - only works for the initial state of the PTA
  - (but can be extended to any state with integer clock values)

- In practice:
  - translation from PTA to MDP can often be done manually
  - (by encoding the PTA directly into the PRISM language)
  - automated translations exist: mcpta and PRISM
  - many case studies, despite “closed” restriction

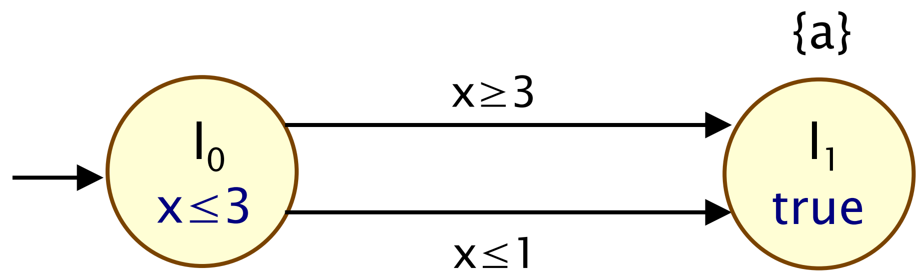
- Problem: can lead to very large MDPs
  - alleviated partially by efficient symbolic model checking

Digital clocks do not preserve PTCTL



- Consider the PTCTL formula $\phi = z.P_{<1}$ [true U $(a \wedge z \leq 1)$]
  - a is an atomic proposition only true in location l_1
- Digital semantics:
  - no state satisfies ϕ since for any state we have $Prob^A(s, \mathcal{E}[z{:=}0],$ true U $(a \wedge z \leq 1)\,) = 1$ for some adversary A
  - hence $P_{<1}$ [true U ϕ] is trivially true in all states

Digital clocks do not preserve PTCTL



- Consider the PTCTL formula $\phi = z.P_{<1}$ [true U $(a \wedge z \leq 1)$]
  - a is an atomic proposition only true in location l_1
- Dense time semantics:
  - any state (l_0, v) where $v(x) \in (1,2)$ satisfies ϕ
    more than one time unit must pass before we can reach l_1
  - hence $P_{<1}$ [true U ϕ] is not true in the initial state

Summary (Part 5)

- Probabilistic timed automata (PTAs)
  - combine probability, nondeterminism, real-time
  - well suited for e.g. for randomised communication protocols
  - MDPs + clocks (or timed automata + discrete probability)

- PTCTL: Temporal logic for properties of PTAs
  - but many useful properties expressible with just reachability

- PTA model checking
  - region graph: decidability results, exponential complexity
  - digital clocks: simple and effective, some scalability issues

- Next: zone-based techniques, abstraction, software