Probabilistic Model Checking

Marta Kwiatkowska
Gethin Norman
Dave Parker

University of Oxford

Part 7 – Probabilistic Timed Automata
Overview

- **Motivation**
- **Time, clocks and zones**
- **Probabilistic timed automata (PTAs)**
  - definition, examples, semantics, time divergence
- **Properties of PTAs: The logic PTCTL**
  - syntax, semantics, examples
- **PTCTL model checking**
  - the region graph
  - forwards and backwards symbolic approaches
  - digital clocks
- **Costs and rewards**
Real-world protocol examples

- Protocols with probability, real-time and nondeterminism

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

- Random choice of waiting time
  - Bluetooth, device discovery phase

- Random choice of a timing delay
  - 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
Time, clocks and clock valuations

- **Dense time domain:** non-negative reals \( \mathbb{R}_{\geq 0} \)

- **Finite set of clocks** \( x \in X \)
  - take values from time domain \( \mathbb{R}_{\geq 0} \), abbreviate to \( \mathbb{R} \)
  - increase at the same rate as real time

- **Clock valuation** \( v \in \mathbb{R}^X \)
  - \( v(x) \) value of clock \( x \)
  - \( v+t \) is time increment for \( v \) with \( t \): \( (v+t)(x) = v(x)+t \) \( \forall x \in X \)
  - \( v[Y:=0] \) clock reset of all clocks in \( Y \subseteq X \)
    - \( v[Y:=0](x)=0 \) if \( x \in Y \)
    - \( v[Y:=0](x)=v(x) \) otherwise
Zones (clock constraints)

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

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

where $x,y \in X$, $c,d \in \mathbb{N}$

– derived logical connectives: $\zeta_1 \lor \zeta_2 = \neg (\neg \zeta_1 \land \neg \zeta_2)$, $\zeta_1 \lor \zeta_2 \rightarrow \ldots$
– get strict inequalities through negation $x > 5 = \neg (x \leq 5) \ldots$

• Closed: do not feature negation (no strict inequalities)

• Diagonal–free: do not feature $x + c \leq y + d$ (no comparisons between clocks)
Zones and clock valuations

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

- Semantics of a zone is the set of clock valuations which satisfy the zone (subset of $\mathbb{R}^N$ if $N$ clocks)
  - more than one zone may have the same semantics:
    $$(x \leq 2) \land (y \leq 1) \land (x \leq y + 2) \text{ and } (x \leq 2) \land (y \leq 1) \land (x \leq y + 3)$$

- Consider only canonical zones
  - 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 $\zeta$, denoted 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 constrains which are greater than $c$
  - for a given $c$, there are only a finite number of c-closed zones
Operations on zones – Set theoretic

- Union of two zones: $\zeta_1 \cup \zeta_2$
Operations on zones – Set theoretic

• Intersection of two zones: $\zeta_1 \cap \zeta_2$
Operations on zones – Set theoretic

- Difference of two zones: $\zeta_1 \setminus \zeta_2$
Operations on zones – clock resets

• $\zeta[X:=0] = \{ v[X:=0] \mid v \gg \zeta \}$
  – clock valuations obtained from $\zeta$ by resetting the clocks in $X$

• $[X:=0]\zeta = \{ v \mid v[X:=0] \gg \zeta \}$
  – clock valuations which are in $\zeta$ if the clocks in $X$ are reset
Operations on zones: $c$–closure

- **c–closure** $\text{close}(\zeta, c)$
  - ignores all constrains which are greater than $c$
Operations on zones: Projection

- Forwards diagonal projection
- \( \nu \zeta = \{ v \mid \exists t \geq 0 . (v-t) \triangleright \zeta \} \)
  - contains the clock valuations that can be reached from \( \zeta \) by letting time pass
Operations on zones: Projection

- Backwards diagonal projection

\[ \zeta = \{ v \mid \exists t \geq 0 . (v + t) \triangleright \zeta \} \]
- contains the clock valuations that, by letting time pass, reach a clock valuation in \( \zeta \)
Overview

- Motivation
- Time, clocks and zones
- **Probabilistic timed automata (PTAs)**
  - definition, examples, semantics, time divergence
- Properties of PTAs: The logic PTCTL
  - syntax, semantics, examples
- PTCTL model checking
  - the region graph
  - forwards and backwards symbolic approaches
  - digital clocks
- Costs and rewards
Probabilistic timed automata – Syntax

- PTA = (Loc, l_{init}, X, \Sigma, inv, prob, L)
  - Loc finite set of locations
  - l_{init} \in Loc the initial location
  - X finite set of clocks
  - \Sigma finite set of events
  - inv : Loc \rightarrow \text{zones}(X) invariant condition
  - prob \subseteq \text{Loc} \times \text{zones}(X) \times \text{dist}(\text{Loc} \times 2^X) probabilistic edge relation
  - L : \text{Loc} \rightarrow \text{AP} labelling function
Probabilistic timed automata – 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
Probabilistic timed automata – Edges

- **Probabilistic edge relation**
  - \( \text{prob} \subseteq \text{Loc} \times \text{zones}(X) \times \Sigma \times \text{dist} \left( \text{Loc} \times 2^X \right) \)

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

- **Edge** \((l,g,\sigma,p,l',X) \subseteq \text{Loc} \times \text{zones}(X) \times \Sigma \times \text{dist} \left( \text{Loc} \times 2^X \right) \times \text{Loc} \times 2^X\)
  - \((l,g,\sigma,p)\) is a probabilistic edge and \(p(l',X)>0\)
  - \(l\) is the source location, \(g\) is the guard, \(\sigma\) is the event
  - \(l'\) is target location
  - \(X\) is the set of clocks to be reset
Probabilistic timed automata – Behaviour

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

- **Start in the initial location with all clocks initialized to zero**
  - let $0$ denote the clock valuation where all clocks have value 0

- **For any state** $(l,v)$ **there is non-deterministic choice between making a discrete transition and letting time pass**
  - **discrete transition** $(l,g,\sigma,p)$ enabled if $g \triangleright \zeta$ and probability of moving to location $l'$ and resetting the clocks $X$ equals $p(l',X)$
  - **time transition** available only if invariant $\text{inv}(l)$ is continuously satisfied while time elapses
Probabilistic timed automata – Example

(di, x=0)
   1.1
   (di, x=1.1)
      0.9
      send
      0.1
      (sr, x=0) (si, x=0)
        8.66
        2.7
        (sr, x=8.66) (si, x=2.7)
          ⋮
          0.95
          retry
          0.05
          (sr, x=0) (si, x=0)
            ⋮
            ⋮

(di, x=0)
   (di, x=0)

sr

true

x≥2
send

x:=0
0.9
0.95
retry

x≥1

x:=0
0.1

x:=0

x≤3
x:=0
0.05

⋯
Probabilistic timed automata – Semantics

Infinite Markov decision process $M_{PTA} = (S_{PTA}, s_{init}, \text{Steps}, L_{PTA})$

- $S_{PTA} \subseteq \text{Loc} \times \mathbb{R}^X$ where $(l,v) \in S_{PTA}$ if and only if $v \triangleright inv(l)$
- $s_{init} = (l_{init}, 0)$

- $\text{Steps} : S_{PTA} \rightarrow 2^{(\Sigma \cup \mathbb{R}) \times \text{Dist}(S)}$ where $((l,v), a, \mu) \in \text{Steps}$ if and only
  - time transition $a = t \geq 0$, $\mu(l,v+t) = 1$ and $v+t' \triangleright inv(l)$ for all $t' \leq t$
  - discrete transition $a = \sigma$, there exists $(l,g,\sigma,p) \in \text{prob}$ such that
    1. $v \triangleright g$
    2. for any $(l',v') \in S_{PTA}$: $\mu(l', v') = \sum_{Y \subseteq X \wedge v'[Y:=0]=v'} p(l', Y)$

- $L_{PTA}(l,v) = L(l)$

actions of $M_{PTA}$ are the events of PTA and non-negative reals ($\Sigma \cup \mathbb{R}_{\geq 0}$)

summation as multiple resets may give same clock valuation (e.g. resetting a clock that equals 0)
Time divergence

• **Restrict 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 during
  – also called *non-zeno* behaviour

• **A path of** $M_{PTA}$ **of the form:** $\omega = s_0(a_1,\mu_1) \ s_0(a_1,\mu_1) \ s_2(a_2,\mu_2)\ldots$
  – where $a_i \in \Sigma \cup \mathbb{R}_{\geq 0}$
  – *duration* up until the $(n+1)$th state

  $$D_\omega(n+1) = \Sigma \{ | a_i | \ 1 \leq i \leq n \ \land \ a_i \in \mathbb{R}_{\geq 0} | \}$$

• **A path** $\omega$ **is time divergent if for any** $t \in \mathbb{R}_{\geq 0}$:
  – there exists $j \in \mathbb{N}$ such that $D_\omega(j) > t$
Time divergence

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

• **Probabilistic divergence motivation** by following example
  - any adversary has a non–divergent path:
    - remain in $l_{\text{init}}$ and do not let 1 time unit elapse
  - chance of such behaviour is 0

![Diagram](image)

**Strong notion** – all paths divergent would mean **NO** divergent adversaries for this example
Overview

• Motivation
• Time, clocks and zones
• Probabilistic timed automata (PTAs)
  – definition, examples, semantics, time divergence
• Properties of PTAs: The logic PTCTL
  – syntax, semantics, examples
• PTCTL model checking
  – the region graph
  – forwards and backwards symbolic approaches
  – digital clocks
• Costs and rewards
PTCTL – Syntax

- Z – set of formula clocks

- $\phi \ ::= \ true \mid a \mid \zeta \mid z. \phi \mid \phi \land \phi \mid \neg \phi \mid P_{\sim p}[\phi \cup \phi]$

- $\phi \cup \phi$ is true with probability $\sim p$

- where a an atomic proposition, $\zeta \in \text{zones}(X \cup Z)$, $z \in Z$ and $p \in [0,1]$, $\sim \in \{<,>,\leq,\geq\}$

- derived from PCTL [BdA95] and TCTL [AD94]
PTCTL – Examples

• $z \cdot P_{>0.99} [\text{packet2unsent} \ U \ \text{packet1delivered} \ \land \ (z < 5)]$
  – 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

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

• $z \cdot P_{\leq0.1} [G (\text{failure} \ \lor \ (z \leq 60))]$
  – the system fails after the first 60 time units have elapsed with probability at most 0.01
PTCTL – Semantics

- Let \((l,v) \in S_{PTA}\) and \(\varepsilon \in \mathbb{R}^Z\) be a formula clock valuation.

combined clock valuation of \(v\) and \(\varepsilon\) satisfies \(\zeta\)

- \((l,v),\varepsilon \models a \iff a \in L(l)\)
- \((l,v),\varepsilon \models \zeta \iff v,\varepsilon \triangleright \zeta\)
- \((l,v),\varepsilon \models z.\phi \iff (l,v),\varepsilon [z:=0] \models \phi\)
- \((l,v),\varepsilon \models \phi_1 \land \phi_2 \iff (l,v),\varepsilon \models \phi_1\) and \((l,v),\varepsilon \models \phi_2\)
- \((l,v),\varepsilon \models \neg \phi \iff (l,v),\varepsilon \models \phi\) is false
- \((l,v),\varepsilon \models P_{\sim p}[\psi] \iff \Pr_{(l,v)}^A\{ \omega \in \text{Path}^A(l,v) \mid \omega,\varepsilon \models \psi\} \sim p\) for all \(A\)

the probability of a path satisfying \(\psi\) meets \(\sim p\) for all divergent adversaries

after resetting \(z\), \(\phi\) is satisfied
PTCTL – Semantics of until

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

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

• Motivation
• Time, clocks and zones
• Probabilistic timed automata (PTAs)
  – definition, examples, semantics, time divergence
• Properties of PTAs: The logic PTCTL
  – syntax, semantics, examples
• **PTCTL model checking**
  – the region graph
  – forwards and backwards symbolic approaches
  – digital clocks
• Costs and rewards
The region graph

- **Region graph construction for PTAs** [KNSS02]
  - adapt the region graph construction for TAs [ACD93]
  - construction *dependent on PTCTL formula* under study

- **For a PTA and PTCTL formula** $\phi$
  - construct a *time-abstract, finite-state MDP* $R(\phi)$
  - translate PTCTL formula $\phi$ to PCTL (denoted $\Phi$)
  - $\phi$ is preserved via region quotient
  - $\phi$ holds in a state of $M_{PTA}$ if and only if $\Phi$ holds in the corresponding state of $R(\phi)$
  - model check $R(\phi)$ using standard methods for MDPs
The region graph – Clock equivalence

- Construction of region graph based on clock equivalence
  - 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) both $t$ and $t'$ are integers or neither is an integer

- The clock valuations $v$ and $v'$ are clock equivalent ($v \equiv v'$) if:
  - for all $x \in X$ one of the following conditions hold:
    (a) $v(x)$ and $v'(x)$ agree on their integral parts
    (b) $v(x) > c$ and $v'(x) > c$
  - for all $x, y \in X$ one of the following conditions hold:
    (a) $v(x) - v(x')$ and $v'(x) - v'(x')$ agree on their integral parts
    (b) $v(x) - v(x') > c$ and $v'(x) - v'(x') > c$
Region graph – Clock equivalence

- $x = 1 \land y = 2$
- $x < y \land 1 < x < 2 \land 1 < y < 2$
- $x = y \land 0 < x < 1$
- $y = 1 \land 2 < x < 3$
Region graph – Clock equivalence

• Fundamental result: if $v \equiv v'$, then $v \triangleright \zeta \iff v' \triangleright \zeta$
  
  – follows $\alpha \triangleright \zeta$ is well defined (where $\alpha$ equivalence class)

• $\beta$ is the successor class of $\alpha$, written $\text{succ}(\alpha) = \beta$, if
  
  – for each $v \in \alpha$, there exists $t > 0$ such that $(v+t, \mathcal{E}+t) \in \beta$
  and $(v+t', \mathcal{E}+t') \in \alpha \cup \beta$ for all $t' < t$
The region graph

- Region graph MDP \((S_R, (l_{init}, 0), \text{Steps}_R, L_R)\)

- \((l, \alpha) \in S_R\) if \(l\) is a location and \(\alpha\) equivalence class of clock valuations over \(X \cup Z\) such that \(\alpha \triangleright inv(l)\)

- \((\text{succ}, \mu) \in \text{Steps}_R(l, \alpha) \iff \text{succ}(\alpha) \triangleright inv(l)\) and \(\mu(l, \text{succ}(\alpha)) = 1\)

- \((\sigma, \mu) \in \text{Steps}_R(l, \alpha) \iff \exists (l, g, \sigma, p) \in \text{prob}\) such that \(\alpha \triangleright g\) and for any \((l', \beta) \in S_R:\)

  \[
  \mu(l', \beta) = \sum_{Y \subseteq X \land \alpha[Y:=0]=\beta} p(l', Y)
  \]

- \(L_R(l, \alpha) = L(l)\)

action set \(\{\text{succ}\} \cup \Sigma\) (\text{succ} corresponds to time passage)

probabilistic transition function \(\text{Steps}_R: S_R \times 2^{(\{\text{succ}\} \cup \Sigma) \times \text{Dist}(S_R)}\)

summation as multiple resets may give same clock equivalence class
Region graph – Example

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

\[
\text{succ} (\text{di}, x=z=0) \rightarrow (\text{di}, 0<x=z<1) \rightarrow (\text{di}, x=z=1) \rightarrow (\text{di}, 1<x=z<2)
\]

\[
0.9 \quad 0.1
\]

\[
\text{(sr, } x=0 \land z=1) \quad (\text{si, } x=0 \land z=1)
\]

\[
\text{di} \quad \text{sr} \quad \text{true}
\]

\[
\text{x}:=0 \quad \text{x} \geq 1 \quad \text{send} \quad \text{retry} \quad \text{x} \geq 2
\]

\[
0.9 \quad 0.1 \quad 0.95 \quad 0.05 \quad 0.05
\]
Region graph – Model checking

- **Problem**
  - prohibitive complexity (exponential in number of clocks and size of largest constant)
  - not implemented (even for timed automata)

- **Improved approach based on zones instead of regions**
  - symbolic states \((l, \zeta)\) where \(\zeta\) is a zone
  - zones are **unions of regions**

- **Two approaches based on:**
  - forwards reachability [KNSS02]
  - backwards reachability [KNSW07]
Overview

- Motivation
- Time, clocks and zones
- Probabilistic timed automata (PTAs)
  - definition, examples, semantics, time divergence
- Properties of PTAs: The logic PTCTL
  - syntax, semantics, examples
- PTCTL model checking
  - the region graph
  - forwards and backwards symbolic approaches
  - digital clocks
- Costs and rewards
Symbolic model checking

- **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
  - \( \text{dpre}[e](S') \) the states that can reach \( S' \) by traversing the edge \( e \)
  - \( \text{tpre}(S') \) the states that can reach \( S' \) by letting time elapse
Symbolic model checking

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

- **tpost** \((l, \zeta) = (l, \neg\zeta \wedge \text{inv}(l))\)
  - \(\neg\zeta\) can be reached from \(\zeta\) by letting time pass
  - \(\neg\zeta \wedge \text{inv}(l)\) must satisfy the invariant of the location \(l\)

- **tpre** \((l, \zeta) = (l, \neg\zeta \wedge \text{inv}(l))\)
  - \(\neg\zeta\) can reach \(\zeta\) by letting time pass
  - \(\neg\zeta \wedge \text{inv}(l)\) must satisfy the invariant of the location \(l\)
Symbolic model checking

- **Edge** $e = (l, g, \sigma, p, l', X)$
  - $l$ is the source
  - $g$ is the guard
  - $\sigma$ is the event
  - $l'$ is the target
  - $X$ is the clock reset

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

- **dpre** $[e](l', \zeta') = (l, [X:=0]\zeta' \land (g \land \text{inv}(l)))$
  - $[X:=0]\zeta'$ the clocks $X$ were reset
  - $[X:=0]\zeta' \land (g \land \text{inv}(l))$ satisfied guard and invariant of $l$
Symbolic model checking – Forwards

• Based on the operation $\text{post}[e](l,\zeta) = \text{tpost}(d\text{post}[e](l,\zeta))$

  – $(l',v') \in \text{post}[e](l,\zeta)$ 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 $S_F=\{\text{tpost}(l_{init},0)\}$ 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 ($c$ largest constant in the PTA)
Symbolic model checking – Forwards

- **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 \((l, g, \sigma, p)\) of PTA such that for any \((l', \zeta') \in Z:\)

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

summation over all the edges of \((l, g, \sigma, p)\) such that applying \(\text{post}\) to \((l, \zeta)\) leads to the symbolic state \((l', \zeta')\)
Symbolic model checking – Forwards

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

- Suppose \( \text{post}[e_1](l, \zeta) = (l_1, \zeta_1) \) and \( \text{post}[e_2](l, \zeta) = (l_2, \zeta_2) \)
  - where \( e_1 \) and \( e_2 \) from the same probabilistic edge

- By definition of \text{post}
  - there exists \((l, v_i) \in (l, \zeta)\) such that a state in \((l_i, \zeta_i)\) 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 \((l, \zeta)\) for which both edges are enabled
Symbolic model checking – Forwards

- 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 $(l_0, x=y)$
Symbolic model checking – Forwards

- **Main result** [KNSS02]
  - obtain *time-abstract, finite-state MDP* over zones
  - *bound on maximum reachability probabilities* only
  - can model check the MDP using standard methods
  - loss of on-the-fly, must construct MDP first

- **Implementations**
  - **KRONOS** pre-processor into PRISM input language, outputs time-abstract MDP [DKN02]
  - **Explicit**, using **Difference Bound Matrices** (DBMs), to PRISM input language [WK05]
  - **Symbolic**, using **Difference Decision Diagrams** (DDDs), via MTBDD-coded PTA syntax directly to PRISM engine [WK05]
Symbolic model checking – Backwards

• Based on pre as opposed to post

\[ \text{pre}[e](l, \zeta) = \text{dpre}[e](\text{tpre}(l, \zeta)) \]

• Suppose pre[e_1](l_1, \zeta_1') = (l, \zeta_1) and pre[e_2](l_2, \zeta_2') = (l, \zeta_2)
  – where e_1 and e_2 from the same probabilistic edge

• By definition of pre
  – for all \((l, v_i) \in (l, \zeta_i)\), a state in \((l_i, \zeta_i')\) can be reached by traversing the edge \(e_i\) and letting time pass
  – therefore, for any \((l, v)\) in the intersection \((l, \zeta_1 \cap \zeta_2)\)
  \((l_i, \zeta_i')\) can be reached by traversing the edge \(e_i\) and letting time pass for both \(i=1\) and \(i=2\)

• To preserve the probabilistic branching structure
  – use both pre and intersection operations
  – unlike the forwards approach results precise
Symbolic model checking – Backwards

- **Backwards Algorithm for PTCTL model checking**
  - **Input**: PTA, PTCTL property $\phi$
  - **Output**: set of symbolic states $\text{Sat}(\phi)$

- $\text{Sat}(a) := \{ (l, \text{inv}(l)) \mid l \in \text{Loc} \text{ and } a \in L(l) \}$
- $\text{Sat}(\zeta) := \{ (l, \text{inv}(l) \land \zeta) \mid l \in \text{Loc} \}$
- $\text{Sat}(\neg \phi) := \{ (l, \text{inv}(l) \land (\lor (l, \zeta) \in \text{Sat}(\phi) \neg \zeta)) \mid l \in \text{Loc} \}$
- $\text{Sat}(\phi_1 \lor \phi_2) := \text{Sat}(\phi_1) \cup \text{Sat}(\phi_2)$
- $\text{Sat}(z.\phi) := \{ (l, [z:=0]\zeta) \mid (l, \zeta) \in \text{Sat}(\phi) \}$
- $\text{Sat}(P_{\sim p}[\phi_1 U \phi_2]) := ?$
Symbolic model checking – Backwards

- Remains to compute the set of states $\text{Sat}(P_{\sim \mathbb{P}}[\phi_1 \cup \phi_2])$
  - sufficient to consider maximum or minimum probability

- Recall from the MDP lecture
  - if $\sim \in \{<,\leq\}$, then $s,\varepsilon \models P_{\sim \mathbb{P}}[\phi_1 \cup \phi_2] \iff p_{\text{max}}(s,\varepsilon, \phi_1 \cup \phi_2) \sim \mathbb{P}$
  - if $\sim \in \{\geq,>\}$, then $s,\varepsilon \models P_{\sim \mathbb{P}}[\phi_1 \cup \phi_2] \iff p_{\text{min}}(s,\varepsilon, \phi_1 \cup \phi_2) \sim \mathbb{P}$

where

$$p_{\text{max}}(s,\varepsilon, \phi_1 \cup \phi_2) = \sup_{A \in \text{Adv}} \Pr_A^s \{\omega \in \text{Path}^A(s) \mid \omega,\varepsilon \models \phi_1 \cup \phi_2\}$$

$$p_{\text{min}}(s,\varepsilon, \phi_1 \cup \phi_2) = \inf_{A \in \text{Adv}} \Pr_A^s \{\omega \in \text{Path}^A(s) \mid \omega,\varepsilon \models \phi_1 \cup \phi_2\}$$
Backwards – Maximum probabilities

- Based on classical backwards exploration for TAs
  - iteratively apply pre operations

- Qualitative case (probability bound 0 or 1)
  - graph based analysis
  - uses methods for finite state MDPs [dA97a, dAKN+00]

- Quantitative case (probability bound in interval (0,1))
  - construct finite–state MDP during backwards exploration
  - states: symbolic states generated during exploration
  - transitions: induced by those of the PTA
  - compute maximal probability for all states of the original PTA through maximum reachability probabilities of the MDP
Backwards – Maximum probabilities

- **Basic algorithm for** $P_{\sim p}[\phi_1 \cup \phi_2]$
  - start with the set of symbolic states $S_B = \text{Sat}(\phi_2)$ then iterate
    for each symbolic state $(l, \zeta) \in S_B$ and edge $e$
      add $\text{pre}[e](l, \zeta)$ to $S_B$
    until set of symbolic states $S_B$ does not change

- Slightly more complicated...

- Restrict to states in $\text{Sat}(\phi_1)$

- Retain the probabilistic branching structure
  - keep track of which symbolic states are constructed through which edges of the PTA and take *conjunctions* of relevant symbolic states
  - relevant symbolic states are those generated by traversing edges taken from the same probabilistic edge
Backwards – Maximum probabilities

• Once the symbolic states $S_B$ have been found

• Construct MDP $(S_B, \text{Steps}_B, L_B)$
  no initial state as we have traversed backwards
  construction similar to forwards approach

• Find maximum probability of reaching $\text{Sat}(\phi_2)$
  – that is compute $p_{\text{max}}(s_B, F a_{\text{Sat}(\phi_2)})$ for all $s_B \in S_B$
  where $a_{\text{Sat}(\phi_2)}$ is an atomic proposition labelling only those
  states in $\text{Sat}(\phi_2)$

• For any state $(l,v)$ of the PTA and formula clock valuation $\mathcal{E}$:
  $p_{\text{max}}((l,v), \mathcal{E}, \phi_1 U \phi_2) = \max \{p_{\text{max}}(s_B, F a_{\text{Sat}(\phi_2)}) \mid (l,v), \mathcal{E} \in S_B \land s_B \in S_B\}$
Backwards – Maximum probabilities

- Maximum probability of reaching $l_4$

$\bullet$ predecessors from the same probabilistic transition: take conjunction of zones

$\begin{align*}
\frac{1}{2} & \quad \frac{1}{2} & \\
(l_1, y \geq x) & (l_1, y = x) & (l_1, x \geq y) \\
\frac{1}{2} & & \frac{1}{2} \\
(l_2, y \geq x) & (l_3, x \geq y) & \\
\frac{1}{2} & & \\
(l_4, \text{true}) & & \\
\end{align*}$
Backwards – Maximum probabilities

- \( z.P_{\sim p}[true \ U sr \land z<4] \) maximum probability of sending the message before 4 time units have passed

\[
\begin{align*}
\text{(si, } 2 \leq x \leq 3 \land z<4) & \xrightarrow{0.95} (sr, z<4) \xrightarrow{0.9} (di, 1 \leq x \leq 2 \land z<4) \\
\text{(si, } 2 \leq x \leq 3 \land z<3) & \xrightarrow{0.05} (di, 1 \leq x \leq 2 \land z<3)
\end{align*}
\]

\[
\begin{align*}
\text{(si, } 2 \leq x \leq 3 \land z<2) & \xrightarrow{0.95} (di, 1 \leq x \leq 2 \land z<4) \\
\text{(si, } 2 \leq x \leq 3 \land z<3) & \xrightarrow{0.05} (di, 1 \leq x \leq 2 \land z<3)
\end{align*}
\]

\[
\begin{align*}
\text{(di, } 1 \leq x \leq 2 \land z<3) & \xrightarrow{0.9} (si, 2 \leq x \leq 3 \land z<4) \\
\text{(di, } 1 \leq x \leq 2 \land z<4) & \xrightarrow{0.9} (si, 2 \leq x \leq 3 \land z<3)
\end{align*}
\]

\[
\begin{align*}
\text{(di, } 1 \leq x \leq 2 \land z<3) & \xrightarrow{0.1} (di, 1 \leq x \leq 2 \land z<4)
\end{align*}
\]

\[
\begin{align*}
\text{(di, } 1 \leq x \leq 2 \land z<4) & \xrightarrow{0.9} (sr, true)
\end{align*}
\]

\[
\begin{align*}
\text{(si, } x \leq 3) & \xrightarrow{0.95} \text{retry}
\end{align*}
\]

for \((l_{\text{init}}, 0), 0\) given by \( p_{\text{max}}((di, 1 \leq di \leq 2 \land z<3), F (sr, z<4)) = 0.995 \)

- \( x := 0 \) – no new symbolic states encountered.

maximum probability of reaching \( sr \land z<4 \) from the initial state corresponds to taking discrete transitions as soon as enabled
Backwards – Minimum probabilities

- Problem: restriction to divergent adversaries
  - minimum probability for until under divergent adversaries
    does not equal minimum under all adversaries

- Example:
  - the minimum probability of formula clock reaching $z > 1$
  - equals 1 under divergent adversaries
  - equals 0 under all adversaries, e.g. consider any adversary
    which lets time converge to a value < 1

- Maximum until probability under divergent adversaries
  does equal maximum under all adversaries
  - just delay time divergence until after satisfaction
Backwards – Minimum probabilities

• Similar problem occurs for timed automata and TCTL

  • $\phi_1 \forall U \phi_2$ – all paths satisfy $\phi_1 U \phi_2$
    – all divergent paths satisfy “true $U z>1$”
    – there exist non-divergent paths not satisfying “true $U z>1$”
    – cannot ignore time divergence when model checking

  • $\phi_1 \exists U \phi_2$ – there exists a path satisfying $\phi_1 U \phi_2$
    – there exists a path satisfying $\phi_1 U \phi_2$ if and only if there exists a divergent path satisfying $\phi_1 U \phi_2$
    – (use same path but let time diverge after $\phi_2$ is reached)
    – can ignore time–divergence when model checking
Backwards – Minimum probabilities

- **Solution for timed automata and TCTL**
  - consider simple case of $\text{AF}\phi$ ($= \text{true} \ \forall U \ \phi$):
  - find state satisfying the dual formula $\text{EG} \neg \phi$
  - (there exists a path for which $\neg \phi$ holds at all times)

- **Compute states satisfying $\text{EG}\phi$ as the greatest fixpoint of**
  \[ H(X) = \phi \land \exists z. (X \exists U z > c) \]
  - 0 iterations: all states
  - 1 iteration: satisfy $\phi$
  - 2 iterations: can satisfy $\phi$ until $c$ time units have passed, ...
  - $k+1$ iterations: can satisfy $\phi$ until $k \cdot c$ time units have passed
  - ... **always** satisfy $\phi$

$c$ is any constant greater than 0
Backwards – Qualitative minimum probabilities

- Set of states satisfying $\neg P_{<1}[G \phi]$ is greatest fixpoint of $H(X) = \phi \land z. \neg P_{<1}[X U (X \lor z>c)]$

  - 0 iterations: all states
  - 1 iteration: all states satisfying $\phi$
  - 2 iterations: all states for which the maximum probability of satisfying $\phi$ until $c$ time units have passed equals 1...
  - $k+1$ iterations: all states for which the maximum probability of satisfying $\phi$ until $k \cdot c$ time units have passed equals 1...
  - ...all states for which the maximum probability of always satisfying $\phi$ equals 1

maximum probability of satisfying $G \phi$ equals 1 (is not less than 1)

maximum probability of satisfying $X U (X \lor z>c)$ equals 1
Backwards – Quantitative minimum probabilities

• For formulae of the form $F \phi$ use the following result

\[
p_{\text{min}}(s, F \phi) = 1 - p_{\text{max}}(s, G \neg \phi)
= 1 - p_{\text{max}}(s, \neg \phi U \neg P_{<1}[G \neg \phi])
\]

and the fact that we have already shown methods for
– computing maximum until probabilities
– the set of states satisfying $\neg P_{<1}[G \phi]$

• Problem reduces to
  – graph analysis (compute $\text{Sat}(\neg P_{<1}[G \phi])$)
  – computation of maximum until probabilities
    (compute $p_{\text{max}}(s, \neg \phi U \neg P_{<1}[G \neg \phi])$)
Backwards – Minimum probabilities

• For formulae of the form $\phi_1 \ U \ \phi_2$ instead use

$$p_{\min}(s, \ \phi_1 \ U \ \phi_2) = 1 - p_{\max}(s, \ \neg\phi_1 \ R \ \neg\phi_2)$$
$$= 1 - p_{\max}(s, \ \neg\phi_2 \ U \ \neg P_{<1}[ \neg\phi_1 \ R \ \neg\phi_2])$$

– operator $R$ (release) is the dual of $U$ (until)
– $\phi_1 \ U \ \phi_2 \equiv \neg (\neg\phi_1 \ R \ \neg\phi_2)$
– $\text{Sat}(\neg P_{<1}[ \neg\phi_1 R \ \neg\phi_2])$ can be computed via a greatest fixpoint
– similar to the method for $\text{Sat}(\neg P_{<1}[ \ G \ \neg\phi])$

• Problem reduces to
  – graph analysis (compute $\text{Sat}(\neg P_{<1}[ \neg\phi_1 R \ \neg\phi_2])$)
  – computation of maximum until probabilities
    (compute $p_{\max}(s, \ \neg\phi_2 \ U \ \neg P_{<1}[ \neg\phi_1 \ R \ \neg\phi_2])$)
Backwards – Minimum probabilities

- $z.P_{\sim p}[F \ sr \land z<6]$ minimum probability of sending the message before 6 time units have passed

  - first step is to find the set of states which satisfy the formula
    \[ \neg P_{<1} [ G \neg (sr \land z<6)] = \neg P_{<1} [ G \ si \lor di \lor (z\geq6)] \]
  - following method described this set is computed as
    \{ (sr,z\geq6), (si,x\leq3 \land z\geq x+3), (di,x\leq2 \land z\geq x+3) \}
  - now find maximum probability of reaching this set of states while remaining in $\neg (sr \land z<6)$
  - i.e. compute $p_{\text{max}}(s, \neg \phi U \neg P_{<1} [ G \neg \phi])$
Backwards – Minimum probabilities

• find maximum probability of reaching
  – \((sr, z \geq 6)\), \((si, x \leq 3 \land z \geq x + 3)\), \((di, x \leq 2 \land z \geq x + 4)\)
  – while remaining in \(\neg (sr \land z < 6)\)

\[(si, x \leq 3 \land z \geq x + 1), (sr, z \geq 4), (di, x \leq 2 \land z \geq x + 2)\]

\[
\text{for } (l_{\text{init}}, 0), 0 \text{ given by } p_{\text{max}}((di, 1 \leq di \leq 2), F_{a_{\text{target}}}) = 0.005
\]

minimum probability of reaching \(sr \land z < 6\) from the initial state corresponds to taking transitions as late as possible
Symbolic model checking – Backwards

- **Main result** [KNS01b, KNSW04]
  - obtain time-abstract, finite-state MDP over zones
  - full PTCTL is preserved via quotient
  - conjunctions of zones to preserve probabilistic branching
  - not on-the fly, must construct MDP first

- **Experimental implementation**
  - Implemented in Java, using Difference Bound Matrices (DBMs)
  - Explicit, into PRISM input language

- **Problem: need to consider non-convex zones**
  - represented as unions of convex zones, i.e. lists of DBMs
  - expensive operations
Overview

• Motivation
• Time, clocks and zones
• Probabilistic timed automata (PTAs)
  – definition, examples, semantics, time divergence
• Properties of PTAs: The logic PTCTL
  – syntax, semantics, examples
• PTCTL model checking
  – the region graph
  – forwards and backwards symbolic approaches
  – digital clocks
• Costs and rewards
Model checking – Digital clocks

• Durations can only take integer durations
  – time domain is $\mathbb{N}$ as opposed to $\mathbb{R}_{\geq 0}$

• Restricted to PTAs class of PTAs, zones must be:
  – closed – do not feature strict inequalities
  – diagonal-free – no comparisons between clocks ($x+c \leq y+d$)

• Based on $\epsilon$–digitisation [HMP92]

• Preserves a subset of properties
  – no nested PTCTL properties
  – zones appearing in formulae closed and diagonal free

• Semantics is an MDP with finite state space
  – need only count up to $c_{\text{max}}$ (max constant in PTA and formula)
  – can employ model checking algorithms for PCTL against MDPs
Model checking – Digital clocks

\[(di, x=z=0) \xrightarrow{0.9} (di, x=z=1) \xrightarrow{0.1} (di, x=z=2)\]

\[(sr, x=0 \land z=1) \xrightarrow{} (si, x=0 \land z=1) \xrightarrow{} (sr, x=0 \land z=2)\]

\[(si, x=1 \land z=2) \xrightarrow{} (si, x=2 \land z=3)\]

\[(sr, x=0 \land z=3) \xrightarrow{0.95} (si, x=0 \land z=3) \xrightarrow{0.05} (si, x=0 \land z=3)\]
Model checking – Digital clocks

- **Main result for digital semantics** [KNPS06]
  - for closed diagonal free PTAs digital semantics preserves minimum/maximum reachability probabilities
  - only for initial state
  
  - extends to formula of the form $z.P_{\sim_p}[\phi_1 U \phi_2]$ if $\phi_1$ and $\phi_2$ contain only atomic propositions and closed diagonal–free zones
  - extends to any state where all clocks have integer values

- **Restriction to closed, diagonal–free** found not to be important for many case studies

- **Problem**: inefficiency for some models, as large constants give rise to very large state spaces
Digital clocks – Probabilistic reachability

- **Probabilistic reachability:**
  - with probability at least 0.999, a data packet is correctly delivered

- **Probabilistic time–bounded reachability**
  - with probability 0.01 or less, a data packet is lost within 5 time units

- **Probabilistic cost–bounded reachability**
  - with probability 0.75 or greater, a data packet is correctly delivered with at most 4 retransmissions

- **Invariance:**
  - with probability 0.875 or greater, the system never aborts

- **Bounded response:**
  - with probability 0.99 or greater, a data packet will always be delivered within 5 time units
Digital clocks – PTCTL not preserved

- Consider the PTCTL formula $\phi = \mathit{z.P} <_1 [\mathit{true U (a_{l1} \land z \leq 1)}]$ 
  - $a_{l1}$ atomic proposition only true in location $l_1$
- Digital semantics:
  - no state satisfies $\phi$ since for any state we have $\text{Prob}^{A}(s,\varepsilon[\mathit{z:=0}], \mathit{true U (a_{l1} \land z \leq 1)}) = 1$ for some adversary $A$
  - hence $P_{<_1}[\mathit{true U \phi}]$ is trivially true in all states
Digital clocks – PTCTL not preserved

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

• Motivation
• Time, clocks and zones
• Probabilistic timed automata (PTAs)
  – definition, examples, semantics, time divergence
• Properties of PTAs: The logic PTCTL
  – syntax, semantics, examples
• PTCTL model checking
  – the region graph
  – forwards and backwards symbolic approaches
  – digital clocks
• Costs and rewards
Costs and rewards

Add reward structure \((\rho, \iota)\) to Probabilistic Timed Automata

- \(\rho : \text{Loc} \to \mathbb{R}_{\geq 0}\) location reward function
  - \(\rho(l)\) is the rate at which the reward is accumulated in location \(l\)
- \(\iota : \Sigma \to \mathbb{R}_{\geq 0}\) event reward function
  - \(\iota(\sigma)\) is the reward associated with performing the event \(\sigma\)

- Generalisation of uniformly priced timed automata

- Special case reward is the elapsed time
  - \(\rho(l) = 1\) for all locations \(l \in \text{Loc}\)
  - \(\iota(\sigma) = 0\) for all events \(\sigma \in \Sigma\)
Expected reachability

- **Expected reward of reaching set of target states**
  - digital clocks semantics preserves expected reachability [KNPS06]
  - can use finite-state MDP algorithm
  - no approach based on zones (yet)

- **Expected reachability properties:**
  - the maximum expected time until a data packet is delivered
  - the minimum expected time until a packet collision occurs
  - the minimum expected number of retransmissions before the message is correctly delivered
  - the minimum expected number of packets sent before failure
  - the maximum expected number of lost messages within the first 200 seconds
Summing up...

- **Probabilistic timed automata (PTAs)**
  - discrete probability distributions only
  - useful in modelling protocols with timing delays and probability
  - extension with continuous distributions exists, but model checking only approximate

- **Implementation**
  - digital clocks via model checking for MDPs
  - forward/backward, experimental implementations only
  - still no satisfactory combination of symbolic probabilistic and real-time data structures

- **More research needed...**
  - contribution to theory and practice