As Cheap as Possible: Efficient Cost-Optimal Reachability for Priced Timed Automata

Paper: Kim Larsen, Gerd Behrmann, Ed Brinksma, Ansgar Fehnker, Thomas Hune, Paul Pettersson and Judi Romijn

Slides: Martin Milata

January 14, 2011

Introduction

Goal of the paper:

- Introduce Linearly Priced Timed Automata.
- Develop zone-based reachability algorithm.
- Use LPTA to solve scheduling problem.
- Another point of view: guided exploration of state-space.

Priced Transition Systems

Definition

Priced transition system is a tuple $(S, s_0, \Sigma, \rightarrow)$, where

- S is a set of states,
- $s_0 \in S$ is initial state,
- Σ is finite set of labels,
- $\rightarrow$ is partial function from $S \times \Sigma \times S$ to $\mathbb{R}_{\geq 0}$ – defines transitions and their costs. Notation: $s \xrightarrow{a}_p s'$.

State space may be infinite – for computing reachability, we need to work with priced symbolic states: (A, π), where $A \subseteq S$, $\pi : A \longrightarrow \mathbb{R}_{\geq 0}$.

Abstract Minimal-Cost Reachability Algorithm

$\textsc{Cost} := \infty$
$\textsc{Passed} := \emptyset$
$\textsc{Waiting} := \{(\{s_0\}, \pi_0)\}$
while $\textsc{Waiting} \neq \emptyset$ **do**
    select (A, π) from $\textsc{Waiting}$
    if $A \cap G \neq \emptyset$ **and** $minCost(A \cap G, \pi) < \textsc{Cost}$ **then**
        $\textsc{Cost} := minCost(A \cap G, \pi)$
    if for all (B, η) in $\textsc{Passed}$: $(B, \eta) \not\sqsubseteq (A, \pi)$ **then**
        add (A, π) to $\textsc{Passed}$
        add $Post_a(A, \pi)$ to $\textsc{Waiting}$ for all $a \in \Sigma$
return $\textsc{Cost}$

Where:

- $(A, \pi) \sqsubseteq (B, \eta) \iff B \subseteq A$ and $\forall s \in B.\ \pi(s) \leq \eta(s)$,
- $minCost(A, \pi) = \inf\{\pi(s) \mid s \in A\}$.

Priced Timed Automata – syntax

Let $\mathcal{B}(\mathbb{C})$ denote the set of conjunctions of atomic constraints of the form $x \sim n$ and $x - y \sim n$, where $\sim \in \{\leq, =, \geq\}$ and $x, y \in \mathbb{C}$.

Definition

Linearly Priced Timed Automaton is a tuple (L, ℓ_0, E, I, P), where

- L is finite set of locations,
- $\ell_0 \in L$ is the initial location,
- $E \subseteq L \times \mathcal{B}(\mathbb{C}) \times 2^{\mathbb{C}} \times L$ are the edges,
- $I : L \longrightarrow \mathcal{B}(\mathbb{C})$ assigns invariants to locations, and
- $P : (L \cup E) \longrightarrow \mathbb{N}$ assigns prices to locations and edges.

Priced Timed Automata – semantics

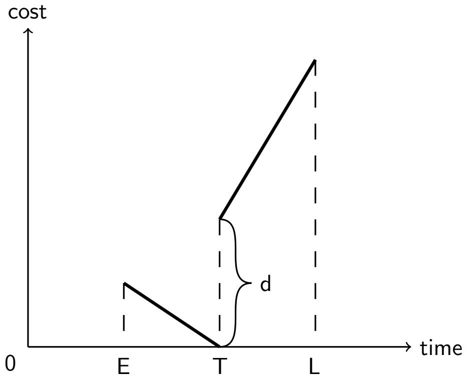
Semantics of PTA (L, ℓ_0, E, I, P) can be given in terms of priced transition system $(S, s_0, \Sigma, \rightarrow)$, where

- $S = L \times \mathbb{R}^{\mathbb{C}}$,
- $s_0 = (\ell_0, u_0)$ – u_0 assigns 0 to every clock,
- $\Sigma = E \cup \{\delta\}$,
- with transitions:

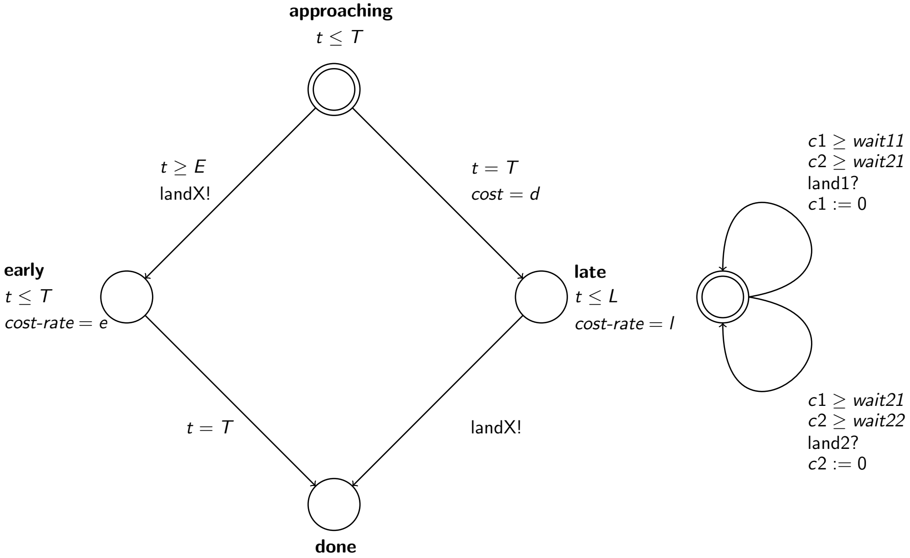
$$(\ell, u) \xrightarrow{\delta}_p (\ell, u+d)$$
$$\text{if } \forall 0 \leq e \leq d.\ u + e \in I(\ell) \text{ and } p = d \cdot P(\ell)$$

$$(\ell, u) \xrightarrow{e}_p (\ell', u')$$
$$\text{if } e = (\ell, g, r, \ell') \in E, u \in g, u' = u[r \mapsto 0], p = P(e)$$

Aircraft landing problem



PTA example


approaching
$t \leq T$
$t \geq E$
landX!
$t = T$
$cost = d$
early
$t \leq T$
$cost\text{-}rate = e$
late
$t \leq L$
$cost\text{-}rate = l$
$t = T$
landX!
done
$c1 \geq wait11$
$c2 \geq wait21$
land1?
$c1 := 0$
$c1 \geq wait21$
$c2 \geq wait22$
land2?
$c2 := 0$

Zones

Symbolic states of the form (ℓ, Z) used in reachability checking – $\ell \in L$, Z is (set of points satisfying) an element of $\mathcal{B}(\mathbb{C})$.

Operations

- $Z^{\uparrow}$ – delay
- $\{r\}Z$ – reset
- $Z \wedge g$ – conjunction with constraint $g \in \mathcal{B}(\mathbb{C})$.

Post-operators in terms of zones:

- $post_{\delta}(\ell, Z) = (\ell, (Z \wedge I(\ell))^{\uparrow} \wedge I(\ell))$,
- $post_{e}(\ell, Z) = (\ell', \{r\}(Z \wedge g))$ for $e = (\ell, g, r, \ell')$.

Extension to sets of priced states?

Priced zones

Priced zone is a triple (Z, c, r), where

- Z is a zone,
- $c \in \mathbb{N}$ is cost of the unique point $\Delta_Z \in Z$ for which $\Delta_Z(x) \leq u(x)$ for all valuations u and clocks x, and
- $r : \mathbb{C} \longrightarrow \mathbb{Z}$ is a function which assigns cost-rate to each clock.

$$Cost(u, Z) = c + \sum_{x \in \mathbb{C}} r(x) \cdot (u(x) - \Delta_Z(x))$$

Priced zones are not directly closed under the *Post*-operations – we may need to split them.

Facets

Let Z be a zone. Whenever $x \sim n$ (resp. $x - y \sim n$) is a constraint of Z, then $Z \wedge x = n$ (resp. $Z \wedge x - y = n$) is a facet.

LF(Z) – lower facets, from constraints of form $x \geq n$
UF(Z) – upper facets, from constraints of form $x \leq n$
LF$_x$(Z) – lower relative facets wrt x: $x \geq m$ and $x - y \geq n$
UF$_x$(Z) – upper relative facets wrt x: $x \leq m$ and $x - y \leq n$

Lemma

- $Z^{\uparrow} = \bigcup_{F \in LF(Z)} F^{\uparrow} = Z \cup \bigcup_{F \in UF(Z)} F^{\uparrow}$,
- $\{y\}Z = \bigcup_{F \in LF_y(Z)} \{y\}F = \bigcup_{F \in UF_y(Z)} \{y\}F$.

Facets

We can define delay and reset for priced constraints.

Definition

Let $\mathcal{Z} = (F, c, r)$ be a priced zone, where F is a lower or upper facet, i.e. $y = n$ is a constraint of F. Let $p \in \mathbb{N}$ be a cost-rate. Then $\mathcal{Z}^{\uparrow p} = (F', c', r')$, where $F' = F^{\uparrow}$, $c' = c$ and $r'(y) = p - \sum_{z \neq y} r(z)$ and $r'(z) = r(z)$ for $z \neq y$.

Definition

Let $\mathcal{Z} = (F, c, r)$ be a priced zone, where F is a relative facet wrt y, i.e. $y - x = m$ is a constraint of F. Then $\{y\}\mathcal{Z} = (F', c', r')$, where $F' = \{y\}F, c' = c$ and $r'(x) = r(y) + r(x)$ and $r'(z) = r(z)$ for $z \neq x$.

Putting it all together

Theorem

Let (L, ℓ_0, E, I, P) be an LPTA. Let $e = (\ell, g, \{y\}, \ell') \in E$ with $P(e) = q, P(\ell) = p, I(\ell) = J$ and let $\mathcal{Z} = (Z, c, r)$ be a priced zone. Then:

$$Post_e(\ell, \mathcal{Z}) = \begin{cases} \{(\ell', \{y\}Q + q) \mid Q \in LF_y(\mathcal{Z} \wedge g)\} & \text{if } r(y) \geq 0 \\ \{(\ell', \{y\}Q + q) \mid Q \in UF_y(\mathcal{Z} \wedge g)\} & \text{if } r(y) \leq 0 \end{cases}$$

$$Post_\delta(\ell, \mathcal{Z}) = \begin{cases} \{(\ell, \mathcal{Z})\} \cup \{(\ell, Q^{\uparrow p} \wedge J) \mid Q \in UF(\mathcal{Z} \wedge J)\} \\ \qquad\qquad \text{if } p \geq \sum_{x \in \mathbb{C}} r(x) \\ \{(\ell, Q^{\uparrow p} \wedge J) \mid Q \in LF(\mathcal{Z} \wedge J)\} \\ \qquad\qquad \text{if } p \leq \sum_{x \in \mathbb{C}} r(x) \end{cases}$$

Computing *minCost* and $\sqsubseteq$ can be reduced to linear programming problem.

Implementation and experiments

- Prototype implementation in UPPAAL.
- Zones represented by Difference Bound Matrices.
- Freely available implementation of simplex algorithm `lp_solve` used to compute *minCost* and $\sqsubseteq$.

- Tested on the aircraft landing problem presented earlier.
- Between 10 and 44 planes, multiple plane types, multiple runways.
- Able to compute all solutions that can be computed by traditional methods, some 50 times faster, some 25 times slower.

Summary & max-plus

- TAs can be extended with the notion of price.
- Zone-based reachability algorithm can be used (with LP solver).
- Computations on priced zones using their facets (splitting).
- PTAs can be successfully used for scheduling problems.

Max-plus polyhedra instead of DBMs?

- MPPs need not have unique Δ_Z (not really a problem).
- Facets can be naturally manipulated in DBMs, which is not the case with internal representation of MPPs, where the principal object is vertex.
- Presented approach is not directly applicable to max-plus polyhedra based reachability checking.