Minimum-Cost Reachability for Priced Timed Automata*

Gerd Behrmann Ansgar Fehnker Thomas Hune
Kim Larsen Paul Pettersson
Judi Romijn Frits Vaandrager

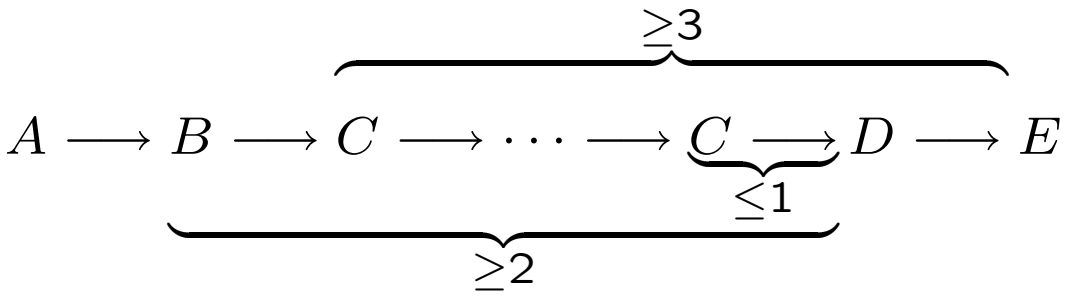
* Supported by ESPRIT project VHS (Verification of Hybrid Systems)

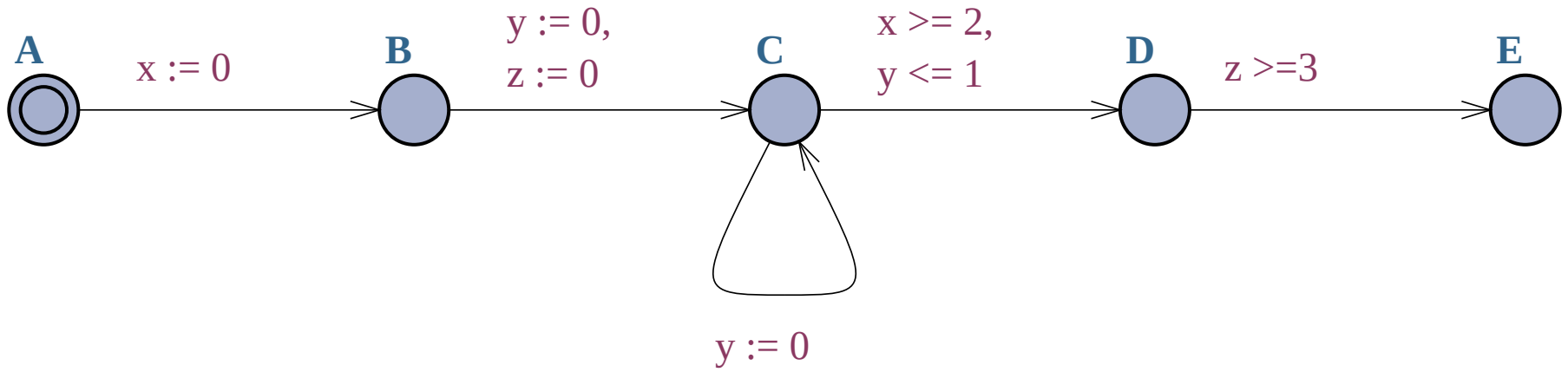
Overview

1. Introduction
2. Linearly Priced Timed Automata
3. Symbolic Semantics
4. Algorithm
5. Conclusions

Introduction

Observation: Many scheduling problems can be phrased naturally as reachability problems for timed automata





Earlier Work

- Asarin & Maler (1999)
  Time optimal control using backward fixed point computation

- VHS consortium (1999)
  Steel plant and chemical batch plant case studies

- Niebert, Tripakis & Yovine (2000)
  Minimum-time reachability using forward reachability analysis

- Behrmann, Fehnker et al (2000)
  Minimum-time reachability using branch-and-bound

Advantages of timed automata approach

- Easy and flexible modeling of systems
- Whole range of verification techniques becomes available
- Controller/program synthesis

Disadvantage

- Existing scheduling approaches perform somewhat better

Our goal

See how far we get; integrate model checking and scheduling theory.

More general cost functions

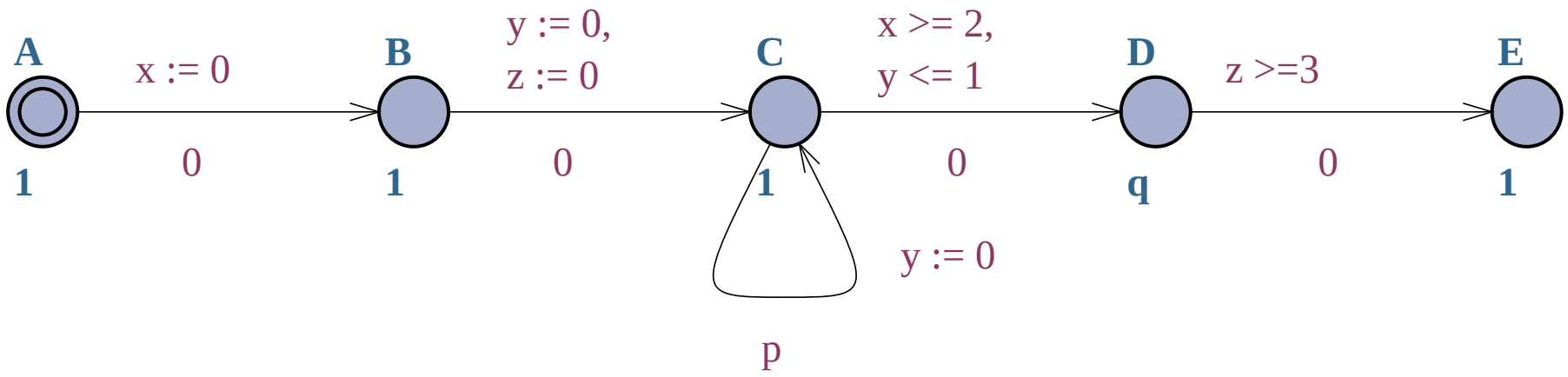
In scheduling theory one is not just interested in shortest schedules; also other cost functions are considered.

This leads us to introducing a model of linearly priced timed automata, which adds prices to locations and transitions.

The price of a transition gives the cost of taking it, and the price of a location specifies the cost per time unit of staying there.

Linearly Priced Timed Automata

Example of linearly priced timed automaton



Optimal cost of reaching E depends on values p and q:

$$\min(3+p, 2+p+q, 2+2q)$$

Definition

A linearly priced timed automaton (LPTA) A over clocks C and actions Act is a tuple (L, l_0, E, I, P) where

- L is a finite set of locations
- l_0 is the initial location
- $E \subseteq L \times \mathcal{B}(C) \times Act \times \mathcal{P}(C) \times L$ is the set of edges
- $I : L \to \mathcal{B}(C)$ assigns invariants to locations
- $P : (L \cup E) \to \mathbb{N}$ assigns prices to locations and edges

If $(l, g, a, r, l') \in E$ then we write $l \xrightarrow{g,a,r} l'$.

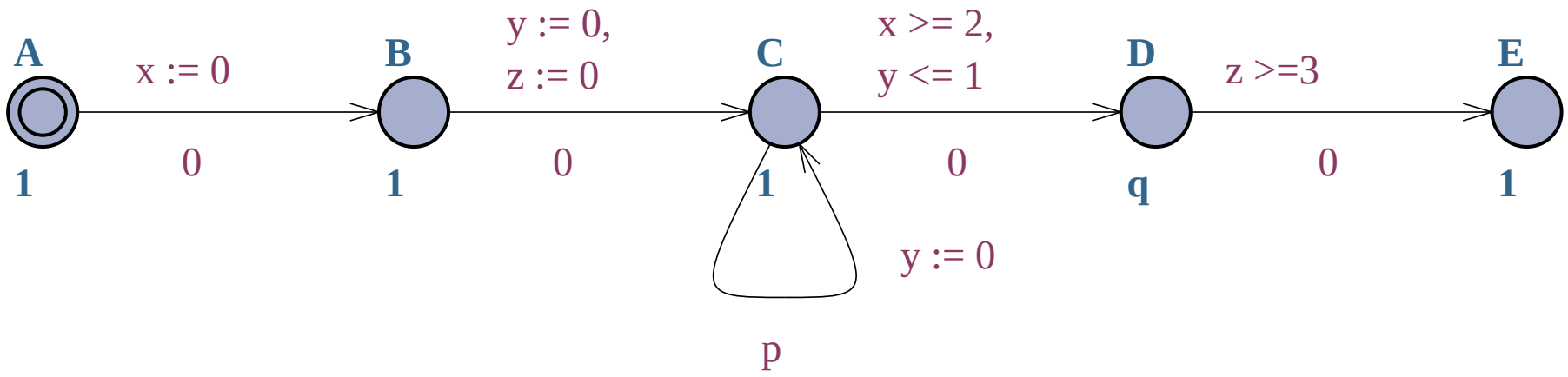
Definition

The semantics of a LPTA A is defined as a transition system with state space $L \times \mathbb{R}^C$, initial state (l_0, u_0) (where u_0 assigns zero to all clocks in C), and transitions:

$$(l,u) \xrightarrow{\epsilon(d),p} (l,u+d) \quad \text{if} \quad u+d \models I(l) \text{ and } p = P(l) * d$$

$$(l,u) \xrightarrow{a,p} (l',u') \quad \text{if} \quad \exists g, r : l \xrightarrow{g,a,r} l',\ u \models g,\ u' = [r \mapsto 0]u,$$
$$u' \models I(l') \text{ and } p = P((l,g,a,r,l'))$$

Example of an execution



$$(A,0,0,0) \xrightarrow{\tau,0} \xrightarrow{\tau,0} \xrightarrow{\epsilon(2),2} (C,2,2,2) \xrightarrow{\tau,p} (C,2,0,2) \xrightarrow{\tau,0} \xrightarrow{\epsilon(1),q} (D,3,1,3) \xrightarrow{\tau,0} (E,3,1,3)$$

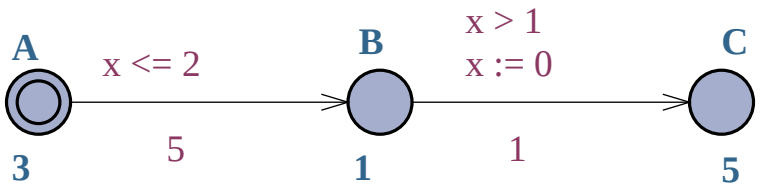
Costs

The cost of a finite execution is the sum of the prices of all the transitions occurring in it.

The minimal cost of a location is the infimum of the costs of the finite executions ending in the location.

The minimum-cost problem for LPTAs is the problem to compute the minimal cost of a given location of a given LPTA.

In the example below, mincost(C) = 7:



Symbolic Semantics

Definition

A priced clock region over a finite set of clocks C

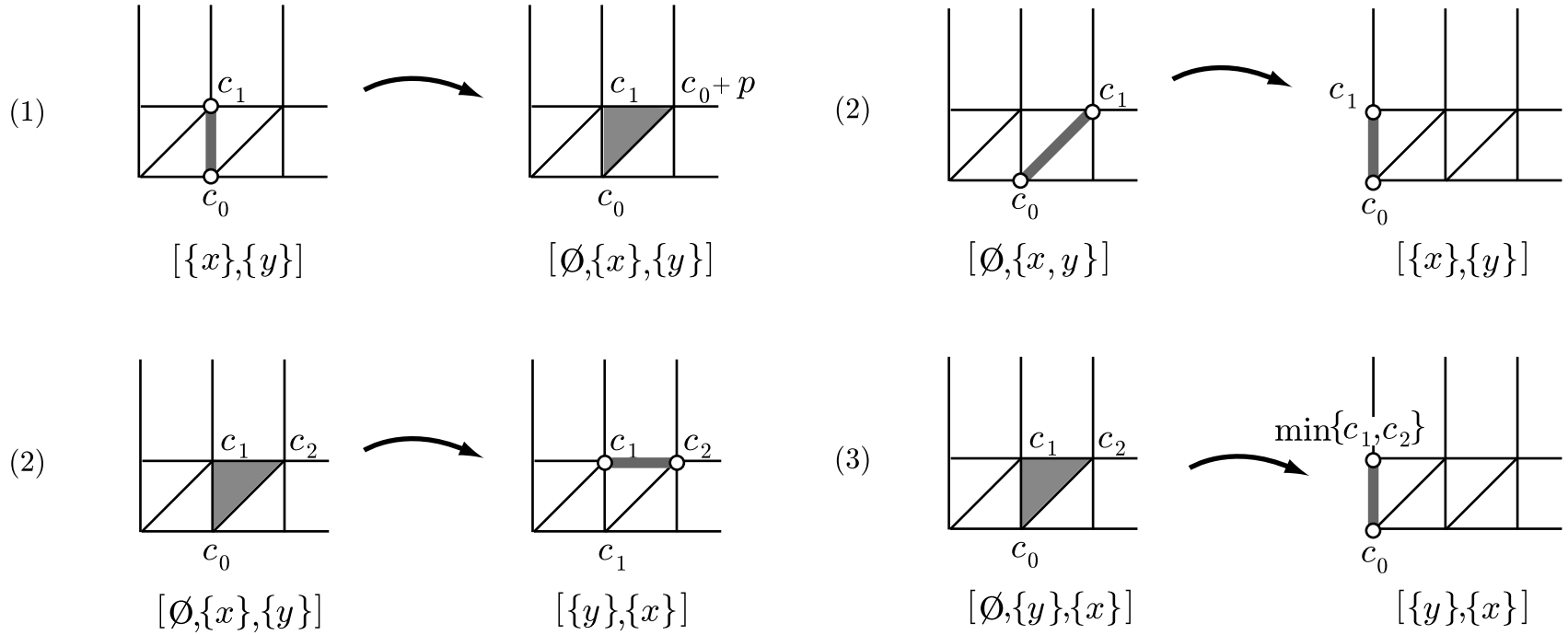
$$R = (h, [r_0, \ldots, r_k], [c_0, \ldots, c_l])$$

is an element of $(C \to \mathbb{N}) \times \mathsf{Seq}(2^C) \times \mathsf{Seq}(\mathbb{N})$, with $k = l$,
$C = \cup_{i \in \{0,\ldots,k\}} r_i$, $i \neq j$ implies $r_i \cap r_j = \emptyset$, and $i > 0$ implies $r_i \neq \emptyset$.

Given a clock valuation $u \in \mathbb{R}^C$ we say that $u \in R$ iff

1. $h(x) = \lfloor u(x) \rfloor$
2. $x \in r_0$ iff $\mathsf{frac}(u(x)) = 0$
3. $x, y \in r_i$ implies $\mathsf{frac}(u(x)) = \mathsf{frac}(u(y))$
4. $x \in r_i$, $y \in r_j$ and $i < j$ implies $\mathsf{frac}(u(x)) < \mathsf{frac}(u(y))$

Examples of delay and reset operations for priced regions



Costs inside regions

For a priced region $R = (h, [r_0, \ldots, r_k], [c_0, \ldots, c_k])$ and clock valuation $u \in R$, the cost of u in R is defined as:

$$\text{cost}(u, R) = c_0 + \sum_{i=0}^{k-1} \text{frac}(u(x_{k-i})) * (c_{i+1} - c_i)$$

where x_j is some clock in r_j.

The minimal cost associated to R is $\text{mincost}(R) = \min(\{c_0, \ldots, c_k\})$.

Symbolic semantics

For priced regions we further define:

- Symbolic transition relation

- Comparison relation $\leq$

Theorem

Let l be a location of a LPTA A. Then

$$\text{mincost}(l) = \min(\{\, \text{mincost}(R) \mid (l, R) \text{ is reachable in symbolic semantics} \,\})$$

Algorithm

Branch-and-bound state-space exploration algorithm

Cost := ∞
Passed := $\emptyset$
Waiting := $\{(l_0, R_0)\}$
while Waiting $\neq \emptyset$ **do**
  select (l, R) from Waiting
  if $l = l_g$ **and** mincost(R) $<$ Cost **then**
    Cost := mincost(R)
  if for all (l, R') in Passed: $R' \not\leq R$ **then**
    add (l, R) to Passed
    for all (l', R') such that $(l, R) \rightarrow (l', R')$: add (l', R') to Waiting
return Cost

Theorem

When the algorithm terminates, the value of Cost equals mincost(l_g).

A LPTA is bounded if there exists a number m such that, for any reachable state (l, u) and for any clock x, $u(x) \leq m$.

Using Higman's Lemma we can prove:

Theorem

The algorithm terminates for any bounded LPTA.

Theorem

The minimum-cost problem for LPTAs can be reduced to the minimum-cost problem for bounded LPTAs.

Theorem

The minimum-cost problem for LPTAs is decidable.

Conclusions

Conclusions

- Basic theory of timed automata has been extended with prices
- Need for efficient (in practice) datastructures and algorithms
- Extension to parametrized setting feasible