IV. Cost or Reward Models

Notizen — 11.03.2005

Examples:

- Mobile Computing: energy is scarce
  ↳ model as costs.

  Possible Questions: minimal cost to reach a desired state?

- Scheduling of a production plant.

  How can you produce all orders cheaper than € x?

Basic Idea: use TA or Markov Chains or...

+ staying in a state: cost proportional to the sojourn time

+ transition: fixed cost.

Lit: As Cheap As Possible / Larsen et al.
CAV 2001. Lecture notes in Computer Science 2102.

Model: Linearly Priced Timed Automata (LPTA)

UPPAAL-Automata + costs.

$$(L, \ell_0, \longrightarrow, I, C, r, i_{tr}, i_{loc}, P)$$

locations

initial location

state labels

clocks

resets

transition invariants

UPPAAL: location invariants

price

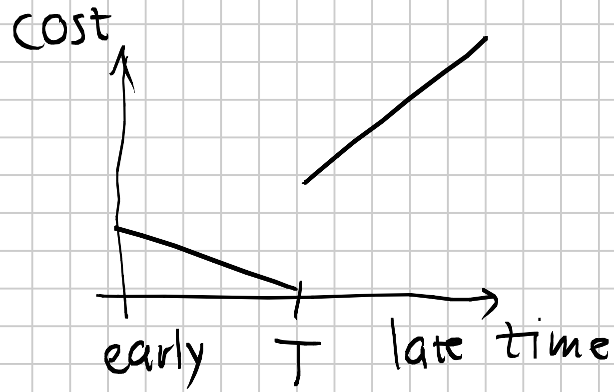
$$P : (L \cup \longrightarrow) \longrightarrow \mathbb{N}$$

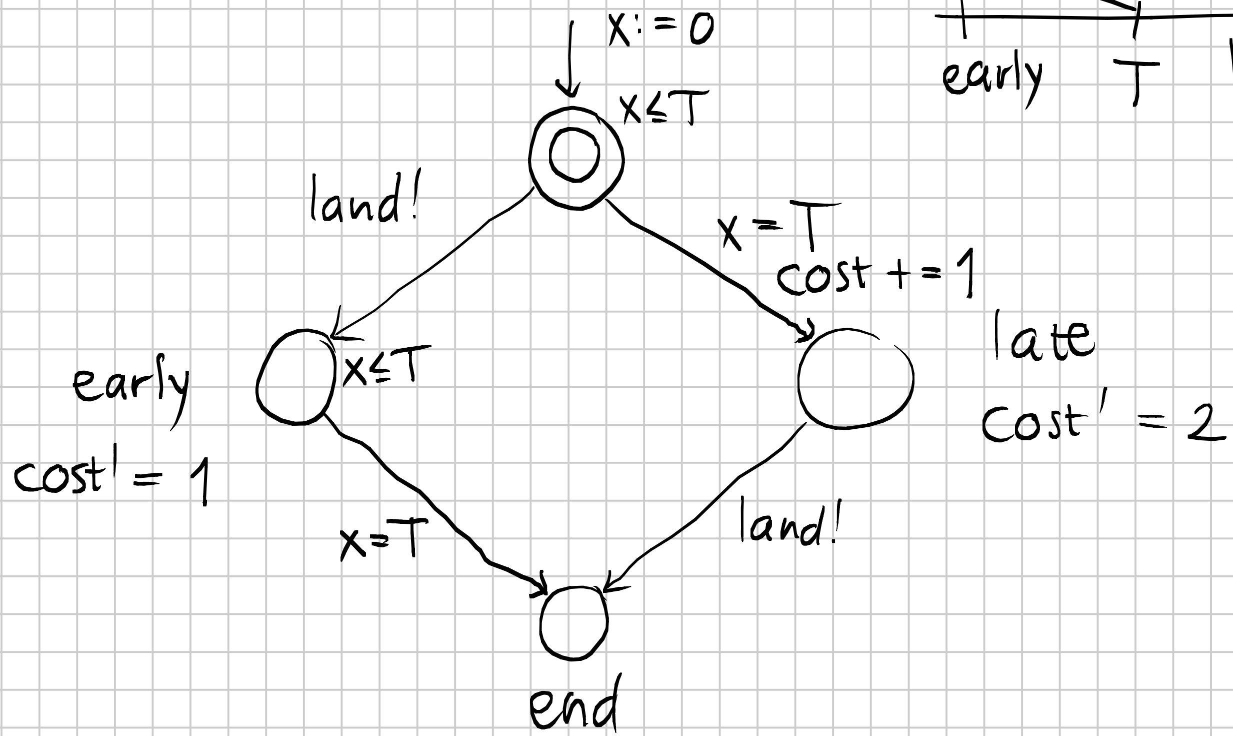
$P(\ell)$ = price rate of location $\ell \in L$

UPPAAL

$P(\ell \xrightarrow{a} \ell')$ = price of a transition

Airplane landing schedule.





States: pairs (location, clock assignment)

State changes: $(\ell, v) \xrightarrow[p]{discrete(a)} (\ell', v')$ if (same as yesterday) and $p = P(\ell \xrightarrow{a} \ell')$

$(\ell, v) \xrightarrow[p]{time(\delta)} (\ell, v')$ if (same procedure as yesterday) and $p = \delta \cdot P(\ell)$

Runs/Paths: same as for TA.

Finite runs: $\sigma = (\ell_0, v_0) \xrightarrow[p_0]{} (\ell_1, v_1) \xrightarrow[p_1]{} \cdots \xrightarrow[p_{n-1}]{} (\ell_n, v_n)$

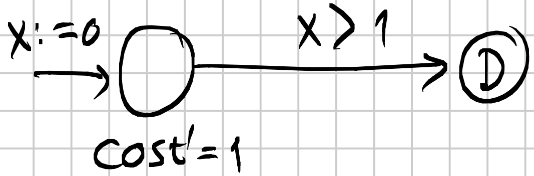
$$P(\sigma) = \sum_{i=0}^{n-1} p_i$$

Question: What is the cheapest way to reach some location in a set $D \subseteq L$ of desired locations?

$$\inf_{\sigma = (\ell_0, 0) \to \cdots \to (\ell_n, v_n)} P(\sigma)$$

ℓ_0: initial location $\qquad$ $\ell_n \in D$

Simplification:

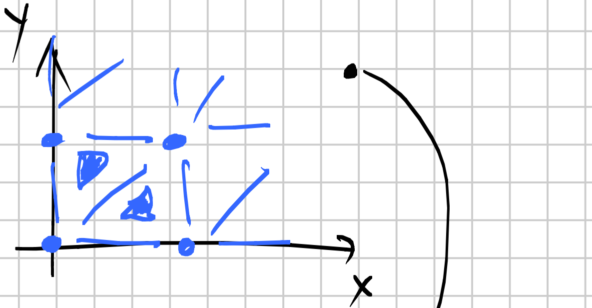


The infimum is not reached in this case.

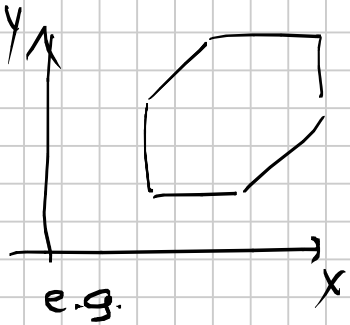
We forbid strict clock constraints.

$x<c,\ x>c$

Zones.



each point represents a clock assgnmt



e.g.

$x \geq 2 \wedge y \geq 3 \wedge x \leq 6 \wedge y \leq 5$

$\wedge (x-y) \leq 2 \wedge (y-x) \leq 1$

simple extension, found in UPPAAL

Symbolic states: pairs (location, zone)

$\xrightarrow{\text{time}(\dots)}$ $\qquad Z\uparrow = \{ v \mid \exists \mu \in Z \ \exists \delta \in \mathbb{R}_0^+ : \mu + \delta = v \}$

$\xrightarrow{\text{discrete}}$: reset. $\quad Z[R:=0] = \{ v \mid \exists \mu \in Z : v(x) = \left(\begin{cases} 0 & \text{if } x \in R \\ \mu(x) \end{cases} \right) \}$

efficient implementations in UPPAAL.

Passing from one symbolic state to another:

$$(\ell, Z) \rightsquigarrow (\ell', Z')$$

$$\ell \xrightarrow{a} \ell' \qquad Z' = \left(Z\!\uparrow \cap\, i_{loc}(\ell) \cap i_{tr}(\ell \xrightarrow{a} \ell')\right)$$
$$[r(\ell \to \ell') := 0] \cap i_{loc}(\ell')$$

Priced zones. $(Z, f: Z \to \mathbb{R}_0^+)$

affine function: $a_0 + a_1 \cdot x + a_2 \cdot y + \cdots$

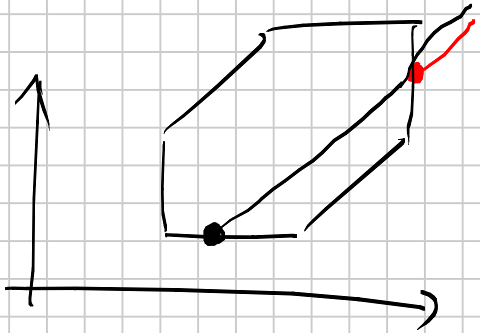
$a_i \in \mathbb{R}$

$x, y \in C$

Priced symbolic states: (ℓ, Z, f)

the set of states (ℓ, v), where $v \in Z$,
with cost $f(v)$

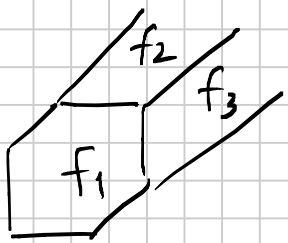
$Z\uparrow$:



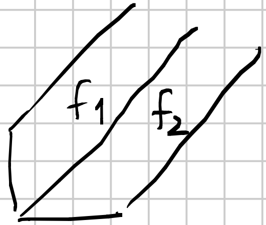
Is the red or black entry cheaper?

Depends on f.

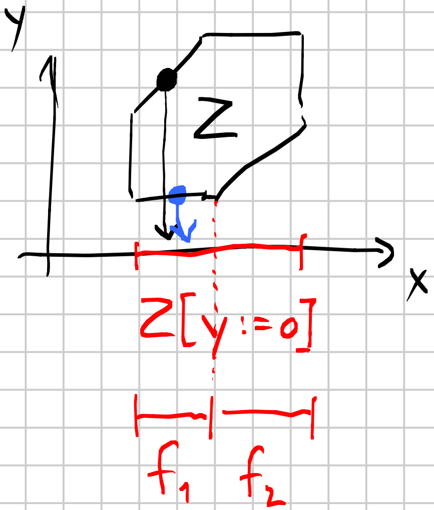
In calculating $Z\uparrow$, we sometimes have to split price functions:



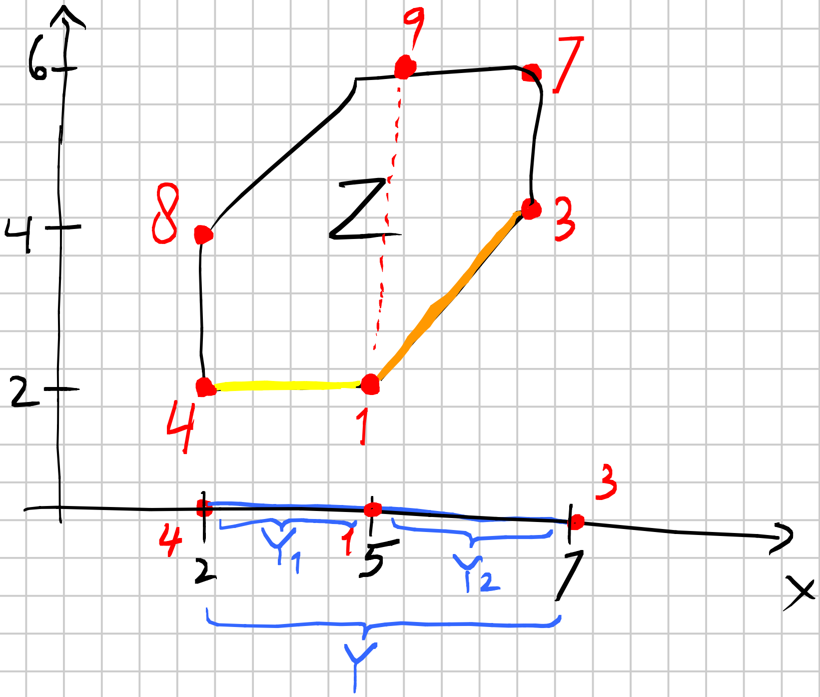
or



$Z[R:=0]$



Example.



$Z = 2 \leq x \leq 7$

$\wedge\ 2 \leq y \leq 6$

$\wedge\ -2 \leq x - y \leq 3$

$f(v) = 2 - v(x) + 2 \cdot v(y)$

$Y = Z[y := 0] = 2 \leq x \leq 7 \ \wedge\ y = 0$

two price functions are needed: $Y_1 = 2 \leq x \leq 5 \wedge y = 0$

has $f_1(v) = 6 - v(x)$

$f_1(2,0) = 4$

$f_1(5,0) = 1$

$Z\uparrow$: similar case analysis.

$Y_2 = (5 \leq x \leq 7 \wedge y = 0)$ has $f_2(v) = \ldots$

Finally,

Global idea of the cost model checking algorithm

1. UPPAAL starts at the initial symbolic state

$$(\ell_0, [x=y=\cdots=0], price=0)$$

2. calculate all reachable priced symbolic states

3. Check whether some state $\in D$ is found.

4. As we tried to spend as little as possible, we can find the cost of the cheapest path to D.

Tool: UPPAAL CORA, more info via UPPAAL website (see related websites).