Synthesis of Optimal Strategies Using HyTech

Patricia Bouyer[1], Franck Cassez[2], Emmanuel Fleury[3] & Kim Guldstrand Larsen[3]

[1] LSV, ENS-Cachan, France [2] IRCCyN, Nantes, France

[3] Comp. Science. Dept., Aalborg University, Danemark

Games in Design and Verification

Boston, Massachusetts, USA

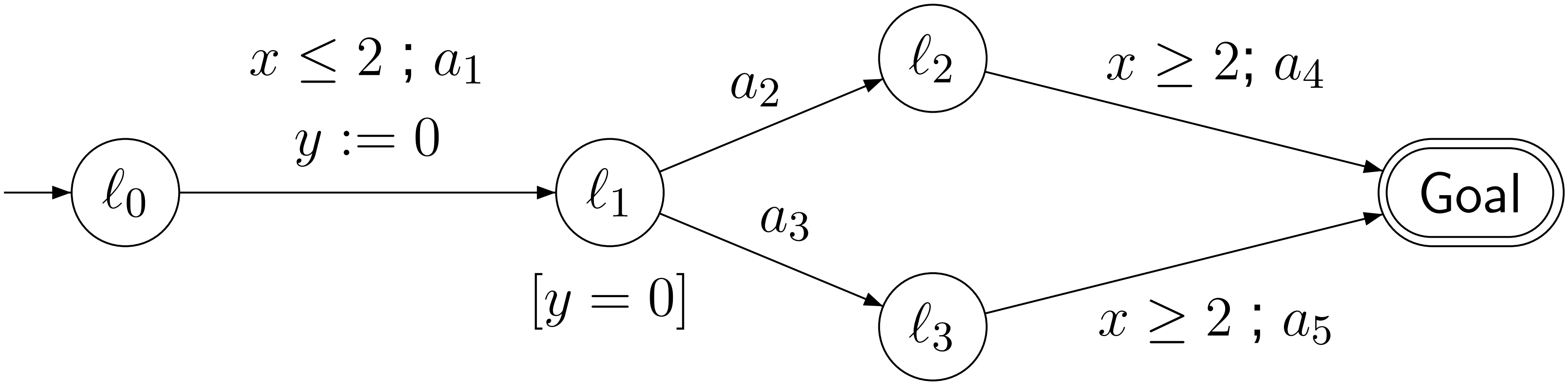
July 18, 2004

http://www.lsv.ens-cachan.fr/aci-cortos/ptga

Contents

Context

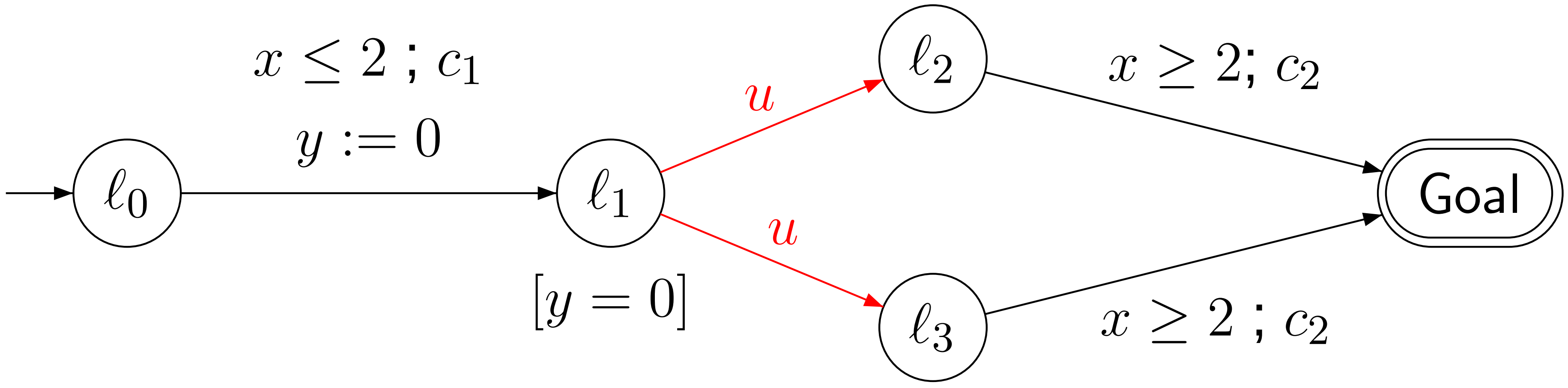
Timed Automata



- Timed Automata + Reachability [AD94]

Context

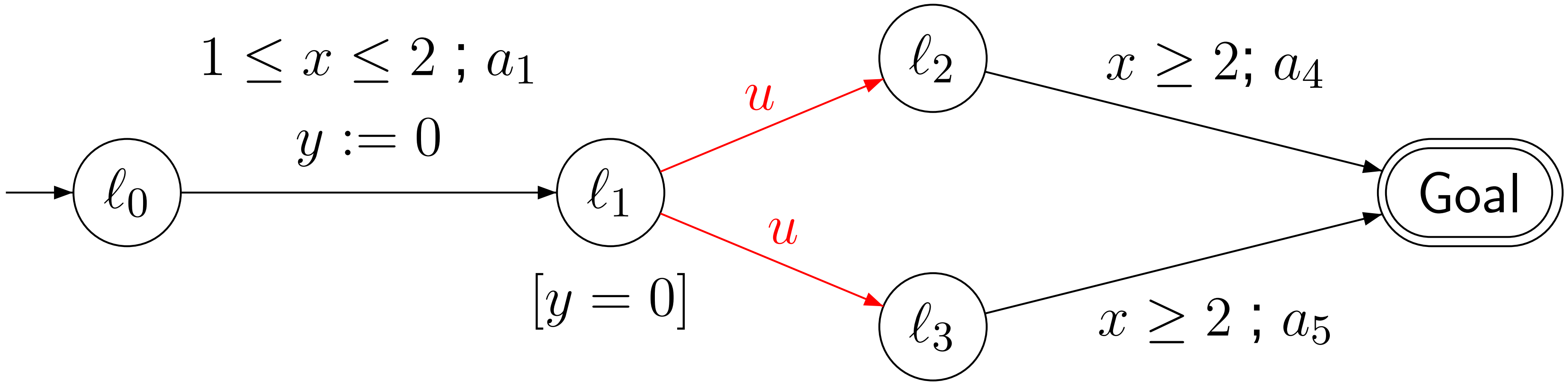
Timed Game Automata



- Timed Automata + Reachability [AD94]
- Timed Game Automata: Control [MPS95, AMPS98]

Context

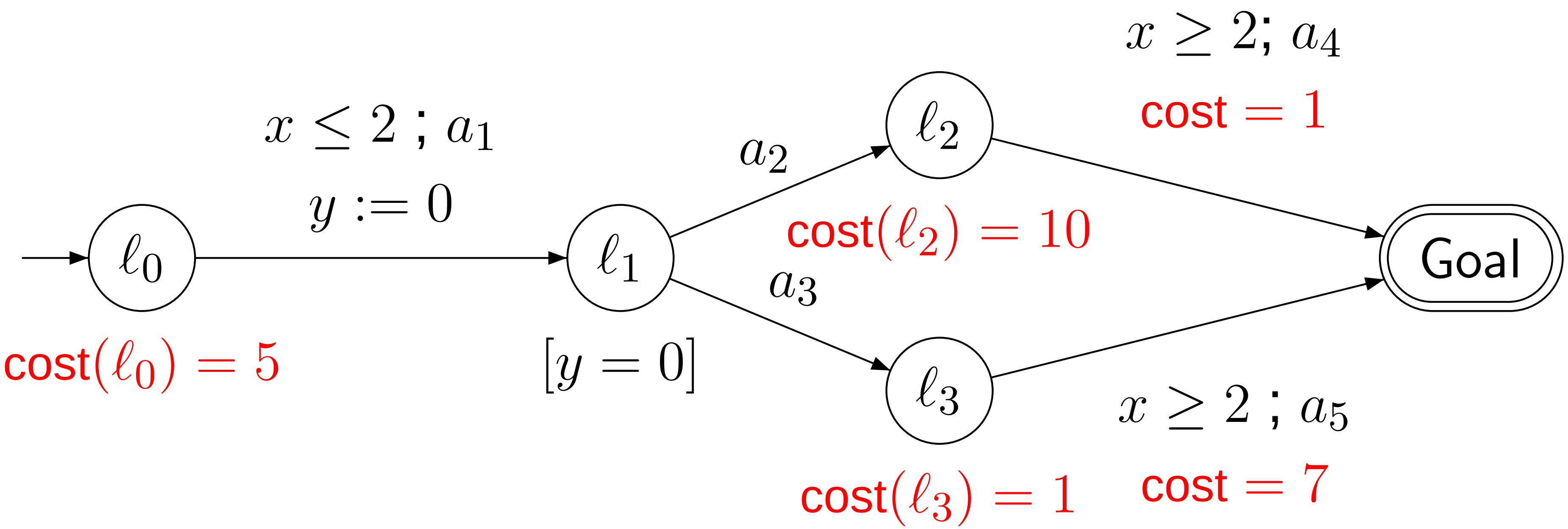
As soon As Possible in Timed Automata



- Timed Automata + Reachability [AD94]
- Timed Game Automata: Control [MPS95, AMPS98]
- Time Optimal Control (Reachability) [AM99]

Context

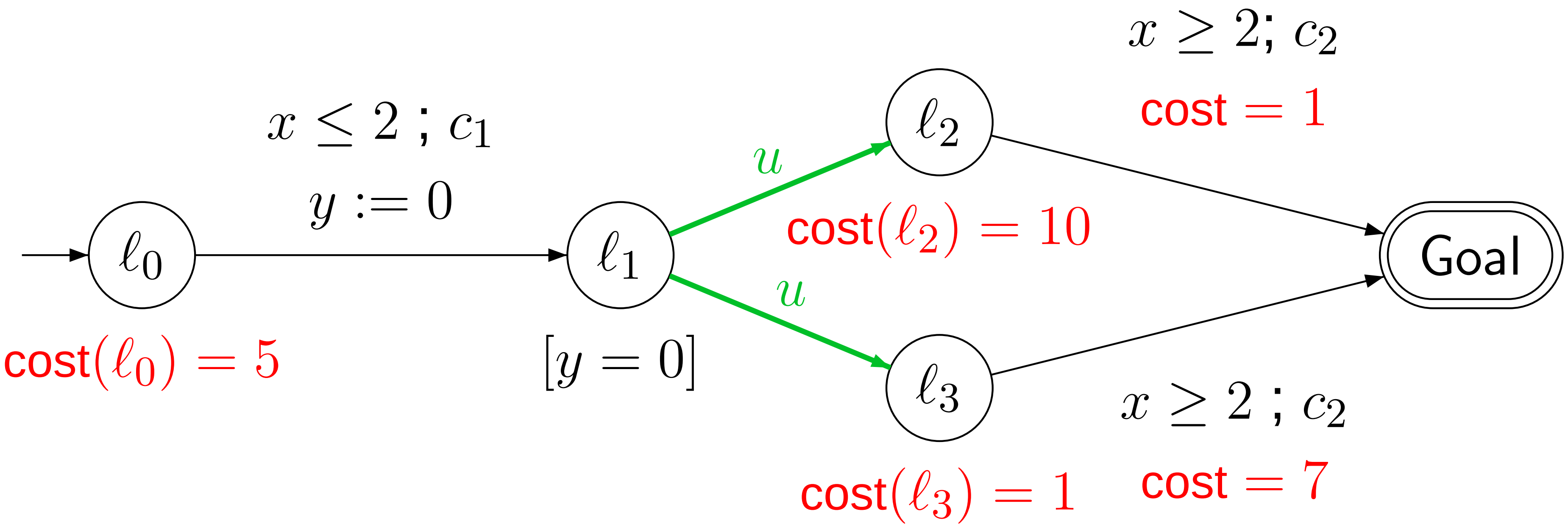
Reachability in Priced Timed Automata



- Timed Automata + Reachability [AD94]
- Timed Game Automata: Control [MPS95, AMPS98]
- Time Optimal Control (Reachability) [AM99]
- Priced (or Weighted) Timed Automata [LBB$^+$01, ALTP01]

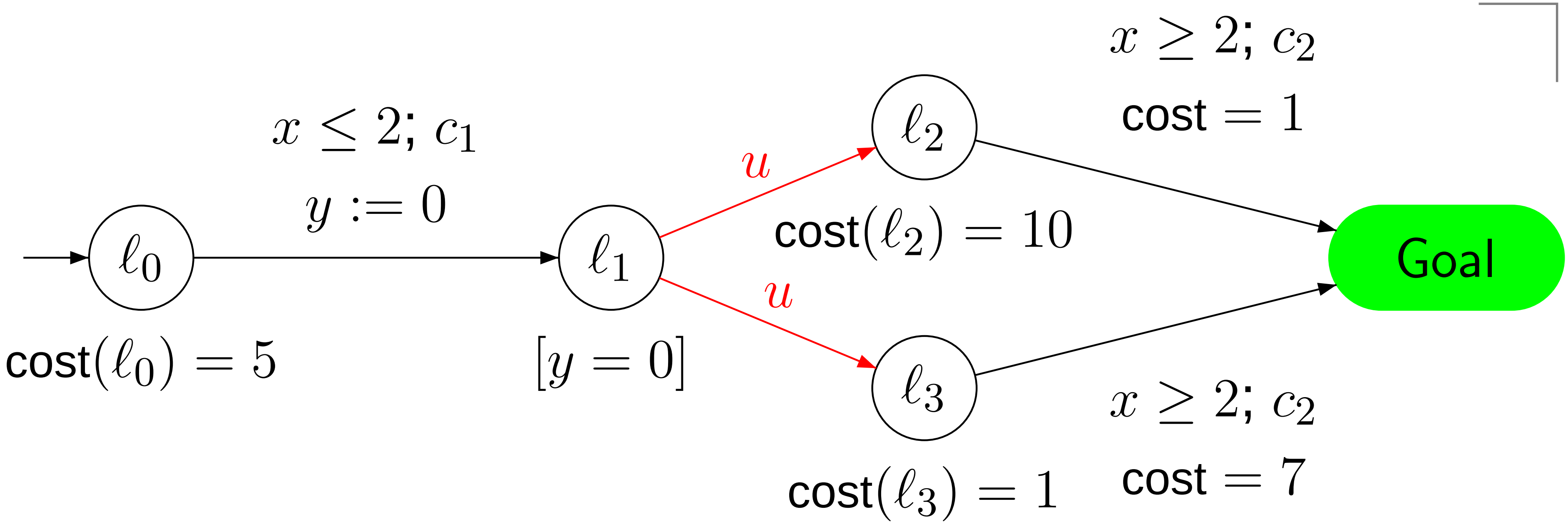
Context

Priced Timed Game Automata



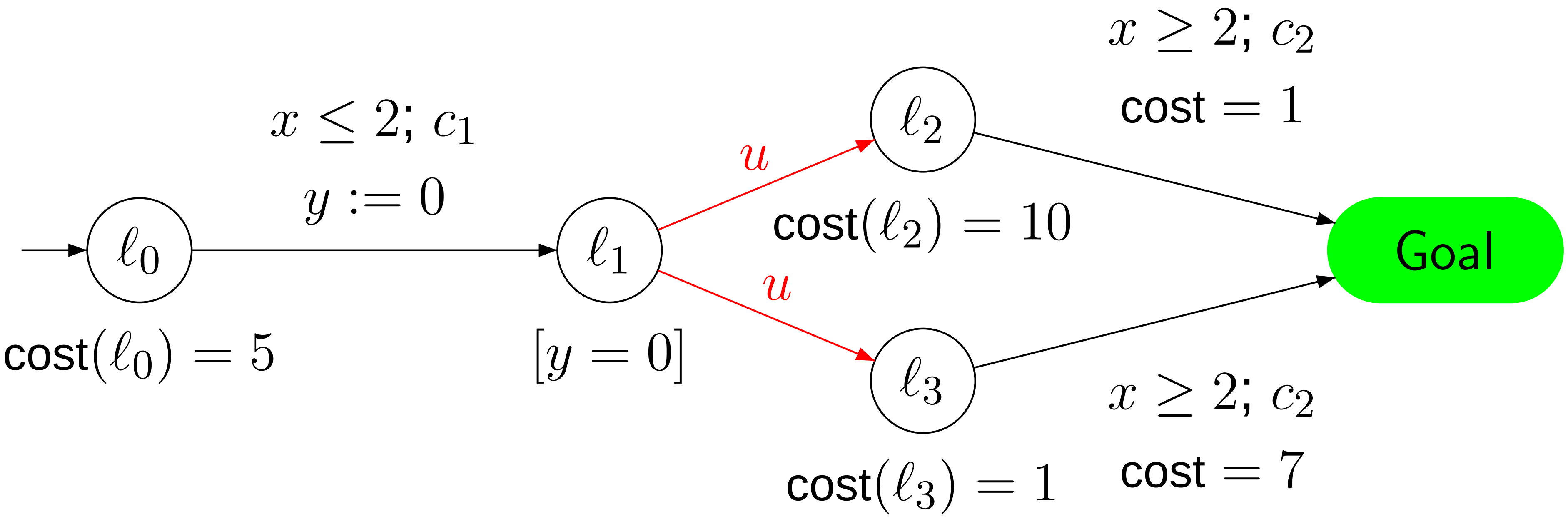
- Timed Automata + Reachability [AD94]
- Timed Game Automata: Control [MPS95, AMPS98]
- Time Optimal Control (Reachability) [AM99]
- Priced (or Weighted) Timed Automata [LBB$^+$01, ALTP01]

A Simple Example



- Model $=$ Game $=$ Controller vs. Environment
- What is the best cost whatever the environment does ?

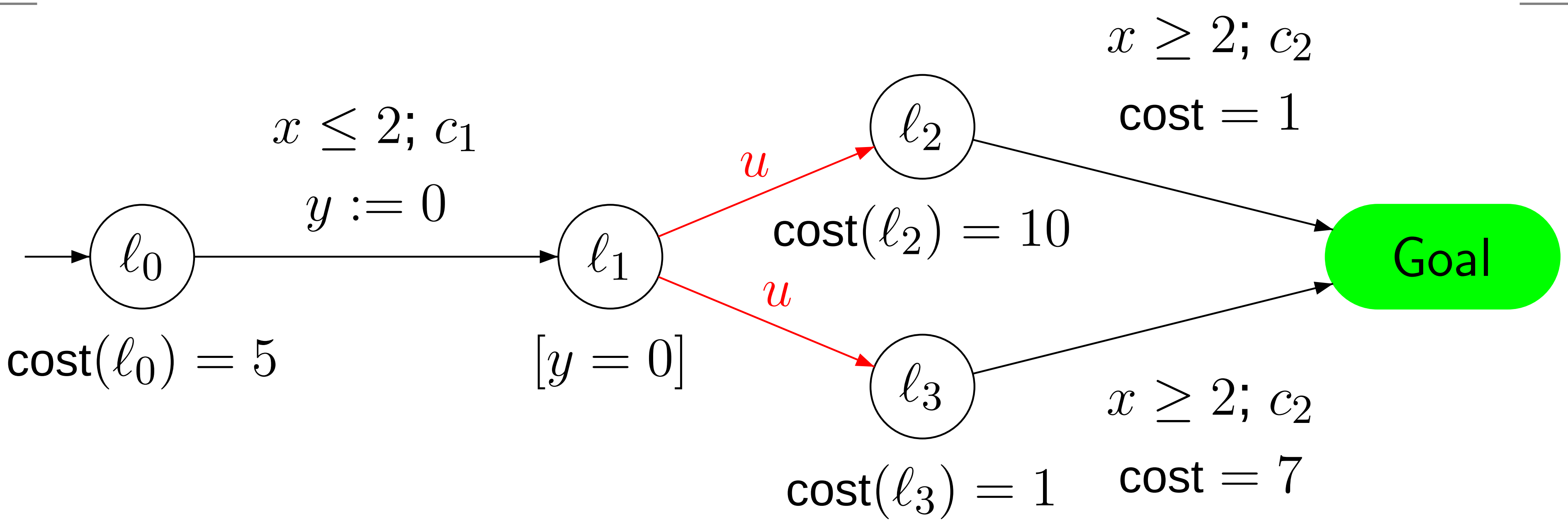
A Simple Example



- What is the best cost whatever the environment does ?

$$\inf_{0 \le t \le 2} \max\{5t + 10(2-t) + 1, 5t + (2-t) + 7\}$$

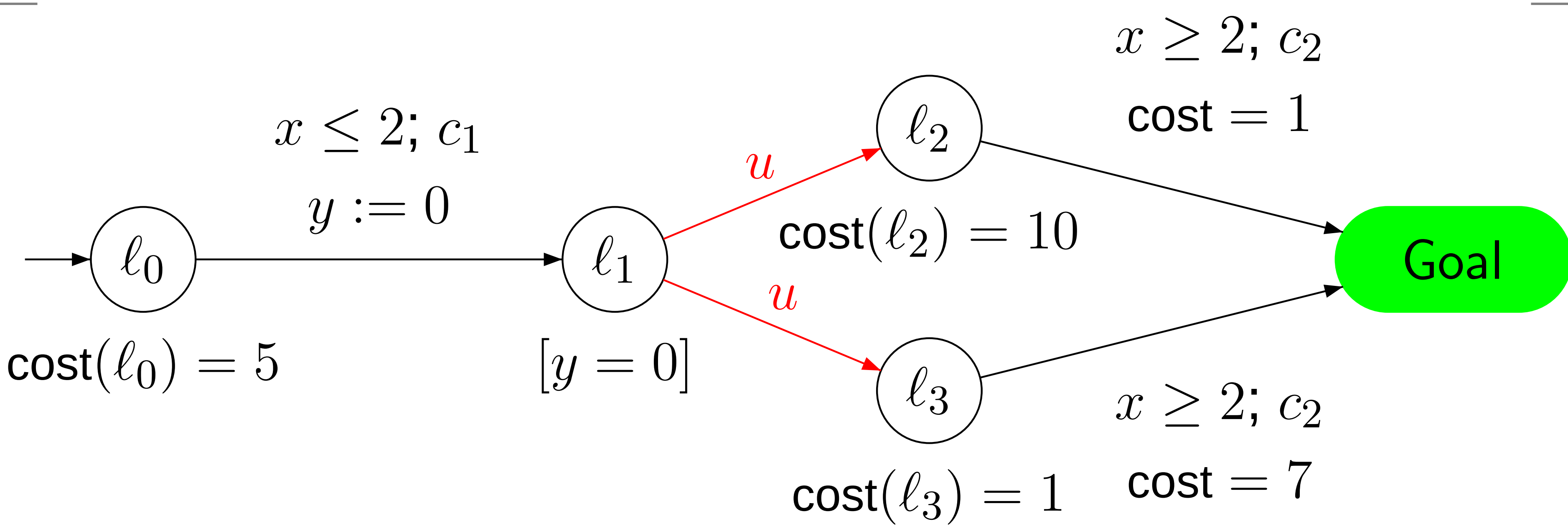
A Simple Example



- What is the best cost whatever the environment does ?

$$\inf_{0 \le t \le 2} \max\{5t + 10(2-t) + 1, 5t + (2-t) + 7\} \text{ at } t = \frac{4}{3} \quad \inf = 14\frac{1}{3}$$

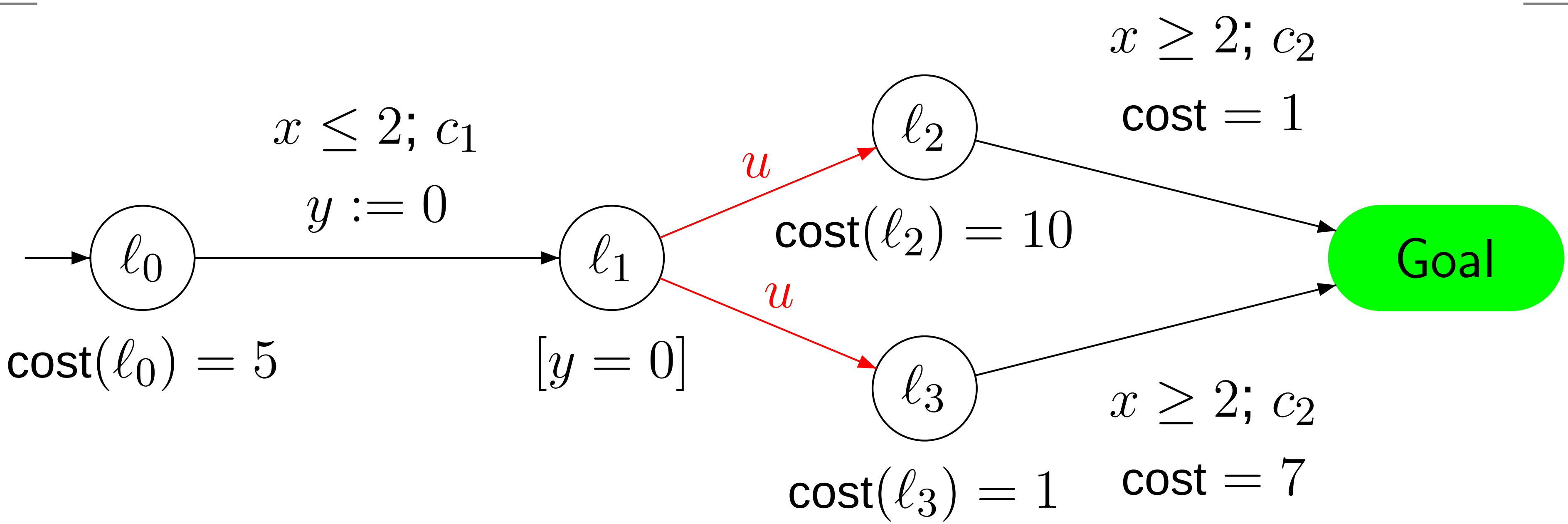
A Simple Example



- What is the best cost whatever the environment does ?

  $\Longrightarrow 14\frac{1}{3}$ at $t = \frac{4}{3}$

A Simple Example



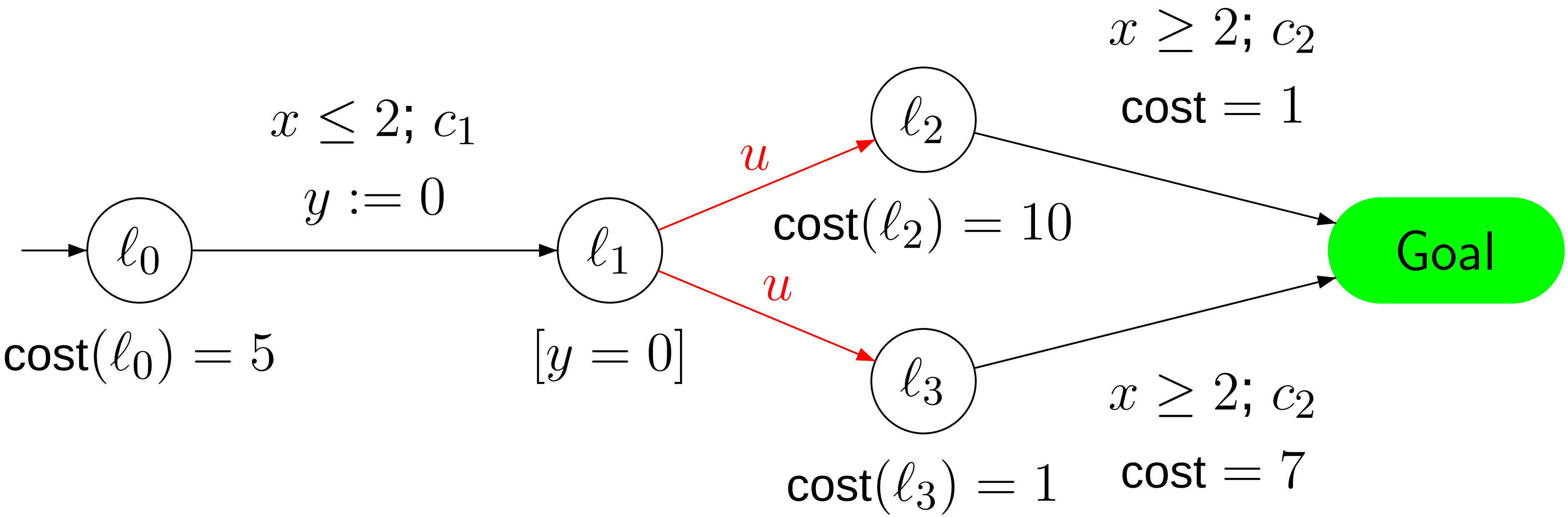
- What is the best cost whatever the environment does ?

  $\Longrightarrow 14\frac{1}{3}$ at $t = \frac{4}{3}$

- Is there a strategy to achieve this optimal cost ?

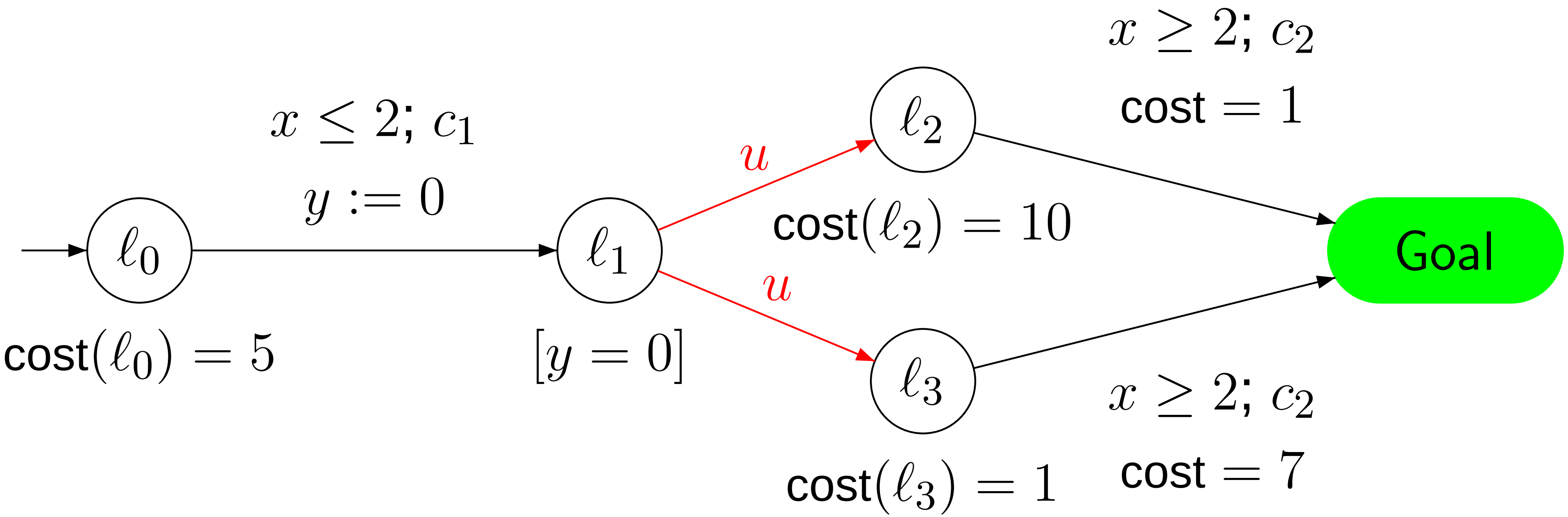
  Yes: because see later

A Simple Example



- What is the best cost whatever the environment does ?
  $\Longrightarrow 14\frac{1}{3}$ at $t = \frac{4}{3}$
- Is there a strategy to achieve this optimal cost ?
  Yes: because see later
- Can we compute such a strategy ?
  Yes: in $\ell_0, x < \frac{4}{3}$ wait then do c_1; in $\ell_{2,3}$ do c_2 when $x \ge 2$

Optimal Control Problems



- Can we find algorithms for these problems on PTGA:
  1. Compute the optimal cost
  2. Decide if there is an optimal strategy
  3. Compute an optimal strategy (if $\exists$)

Related Work

- La Torre et al. [LTMM02] (IFIP TCS'02)
  - Acyclic Priced Timed Game Automata
  - Recursive definition of optimal cost [$\Longrightarrow$ La Torre et al. Def.]
  - Computation of the infimum of the optimal cost
    $\mathrm{OptCost} = 2$ could be 2 or $2 + \varepsilon$
  - No strategy synthesis

Related Work

- La Torre et al. [LTMM02] (IFIP TCS'02)
  Acyclic Games, infimum, no strategy synthesis
- Alur et al. [ABM04] (ICALP'04)
  - bounded optimality: optimal cost within k steps
  - complexity bound: exponential in k and #states of the PTGA
  - no bound for the more general optimal problem
  - Computation of the infimum of the optimal cost
  - no strategy synthesis

Related Work

- La Torre et al. [LTMM02] (IFIP TCS'02)
  Acyclic Games, infimum, no strategy synthesis
- Alur et al. [ABM04] (ICALP'04)
  bounded optimality, complexity bound, infimum, no strategy synthesis
- Our work [BCFL04]:
  - Run-based definition of optimal cost
  - We can decide whether $\exists$ an optimal strategy
  - We can synthesize an optimal strategy (if $\exists$)
  - We can prove structural properties of optimal strategies
  - Applies to Linear Hybrid Game (Automata)

Contents

Priced Timed Game Automata

A Timed Game Automaton (PTGA) G is a tuple $(L, \ell_0, \mathsf{Act}, X, E, \mathsf{inv}, \mathsf{cost})$ where:

- L is a finite set of locations;
- $\ell_0 \in L$ is the initial location;
- $\mathsf{Act} = \mathsf{Act}_c \cup \mathsf{Act}_u$ is the set of actions (partitioned into controllable and uncontrollable actions);
- X is a finite set of real-valued clocks;
- $E \subseteq L \times \mathcal{B}(X) \times \mathsf{Act} \times 2^X \times L$ is a finite set of transitions;
- $\mathsf{inv} : L \longrightarrow \mathcal{B}(X)$ associates to each location its invariant;

Priced Timed Game Automata

A Priced Timed Game Automaton (PTGA) G is a tuple $(L, \ell_0, \mathsf{Act}, X, E, \mathsf{inv}, \mathsf{cost})$ where:

- L is a finite set of locations;
- $E \subseteq L \times \mathcal{B}(X) \times \mathsf{Act} \times 2^X \times L$ is a finite set of transitions;
- Priced Version: $\mathsf{cost} : L \cup E \longrightarrow \mathbb{N}$ associates to each location a cost rate and to each discrete transition a cost value.

[⟹ Example]

Priced Timed Game Automata

A Priced Timed Game Automaton (PTGA) G is a tuple $(L, \ell_0, \mathsf{Act}, X, E, \mathsf{inv}, \mathsf{cost})$ where:

- L is a finite set of locations;
- $E \subseteq L \times \mathcal{B}(X) \times \mathsf{Act} \times 2^X \times L$ is a finite set of transitions;
- Priced Version: $\mathsf{cost} : L \cup E \longrightarrow \mathbb{N}$ associates to each location a cost rate and to each discrete transition a cost value. [$\Longrightarrow$ Example]

- we assume that PTGA are deterministic w.r.t. controllable actions (+ time-deterministic)
- A reachability PTGA (RPTGA) $=$ PTGA with distinguished set of states $\mathsf{Goal} \subseteq L$.

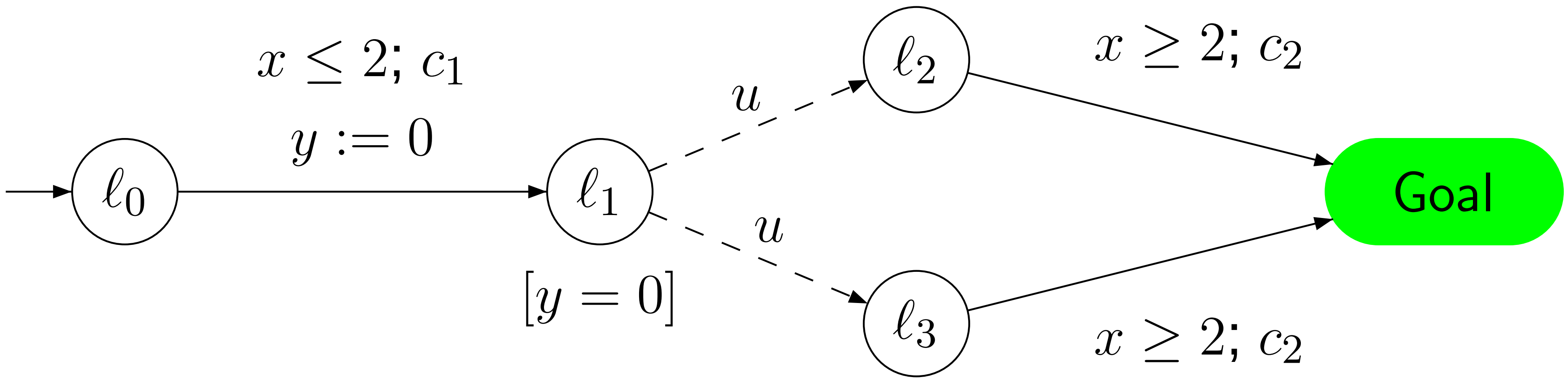
Configurations, Runs, Costs

- configuration: (ℓ, v) in $L \times \mathbb{R}^X_{\geq 0}$
- step: $t_i = (\ell_i, v_i) \xrightarrow{\alpha_i} (\ell_{i+1}, v_{i+1})$
  $$\begin{cases} \alpha_i \in \mathbb{R}_{>0} \implies \ell_{i+1} = \ell_i \wedge v_{i+1} = v_i + \alpha_i \\ \alpha_i \in \mathsf{Act} \implies \exists(\ell_i, g, \alpha_i, Y, \ell_{i+1}) \in E \wedge v_i \models g \ \wedge v_{i+1} = v_i[Y] \end{cases}$$
- run $\rho = t_0 t_1 t_2 \cdots t_{n-1} \cdots$ finite or infinite sequence of t_i
- cost of a transition:
  $$\begin{cases} \mathsf{Cost}(t_i) = \alpha_i.\mathsf{cost}(\ell_i) \text{ if } \alpha_i \in \mathbb{R}_{>0} \\ \mathsf{Cost}(t_i) = \mathsf{cost}((\ell_i, g, \alpha_i, Y, \ell_{i+1})) \text{ if } \alpha_i \in \mathsf{Act} \end{cases}$$
- if ρ finite $\mathsf{Cost}(\rho) = \sum_{0 \leq i \leq n-1} \mathsf{Cost}(t_i)$
- winning run if $\mathsf{States}(\rho) \cap \mathsf{Goal} \neq \emptyset$

Strategies

- strategy f over G = partial function from $\mathsf{Runs}(G)$ to $\mathsf{Act}_c \cup \{\lambda\}$.
- $\mathsf{Outcome}((\ell, v), f)$ (outcomes) of f from configuration (ℓ, v) $=$ a subset of $\mathsf{Runs}((\ell, v), G)$ [$\Longrightarrow$ Formal Definition of Outcome]

Strategies



Example:
$$\begin{cases} f(\ell_0, x < \frac{4}{3}) = \lambda \quad f(\ell_0, \frac{4}{3} \leq x \leq 2) = c_1 \\ f(\ell_1, -) \text{ undefined} \\ f(\ell_2, x < 2) = \lambda \quad f(\ell_2, x \geq 2) = c_2 \\ f(\ell_3, x < 2) = \lambda \quad f(\ell_3, x \geq 2) = c_2 \end{cases}$$

Strategies

- strategy f over G = partial function from $\mathsf{Runs}(G)$ to $\mathsf{Act}_c \cup \{\lambda\}$.
- $\mathsf{Outcome}((\ell, v), f)$ = outcomes of f from configuration (ℓ, v);

[$\Longrightarrow$ Formal Definition of Outcome]

- a strategy f is winning from (ℓ, v) if

$$\mathsf{Outcome}((\ell, v), f) \subseteq \mathsf{WinRuns}((\ell, v), G)$$

- The cost of f from (ℓ, v) is

$$\mathsf{Cost}((\ell, v), f) = \sup\{\mathsf{Cost}(\rho) \mid \rho \in \mathsf{Outcome}((\ell, v), f)\}$$

(Formal) Optimal Control Problems

Optimal Cost Computation Problem: compute the optimal cost one can expect from $s_0 = (\ell_0, \vec{0})$

$$\mathsf{OptCost}(s_0, G) = \inf\{\mathsf{Cost}(s_0, f) \mid f \in \mathsf{WinStrat}(s_0, G)\}$$

Optimal Strategy Existence Problem: determine whether the optimal cost can actually be reached

$$\exists ? f \in \mathsf{WinStrat}(s_0, G) \text{ s.t. } \mathsf{Cost}(s_0, f) = \mathsf{OptCost}(s_0, G)$$

Optimal Strategy Synthesis Problem: in case an optimal strategy exists, compute a witness.

(Formal) Optimal Control Problems

Optimal Cost Computation Problem: compute the optimal cost one can expect from $s_0 = (\ell_0, \vec{0})$

$$\mathsf{OptCost}(s_0, G) = \inf\{\mathsf{Cost}(s_0, f) \mid f \in \mathsf{WinStrat}(s_0, G)\}$$

Optimal Strategy Existence Problem: determine whether the optimal cost can actually be reached

$$\exists ? f \in \mathsf{WinStrat}(s_0, G) \text{ s.t. } \mathsf{Cost}(s_0, f) = \mathsf{OptCost}(s_0, G)$$

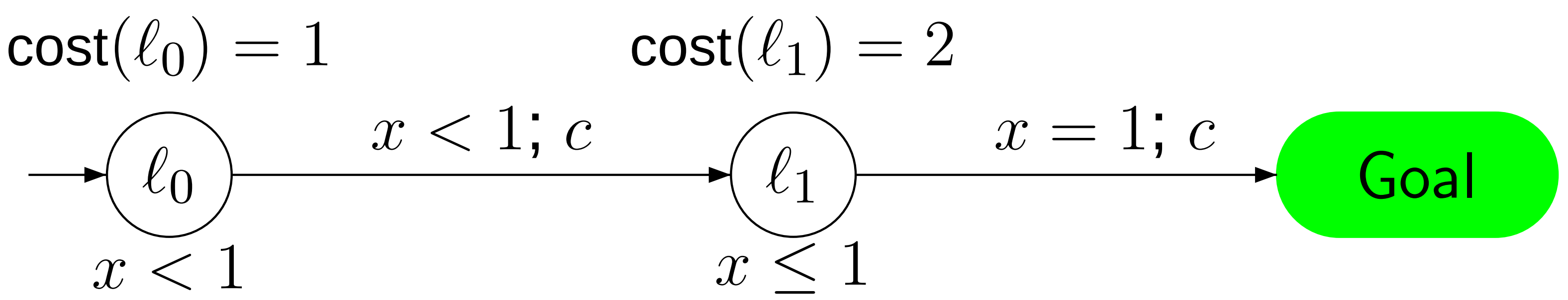
Optimal Strategy Synthesis Problem: in case an optimal strategy exists, compute a witness.

Relation to La Torre et al. work [LTMM02] (acyclic game):

Theorem 1: $\mathsf{OptCost}(s_0, G) = O(s_0)$

[$\Longrightarrow$ Definition of $O(q)$]

Example: No Optimal Strategy



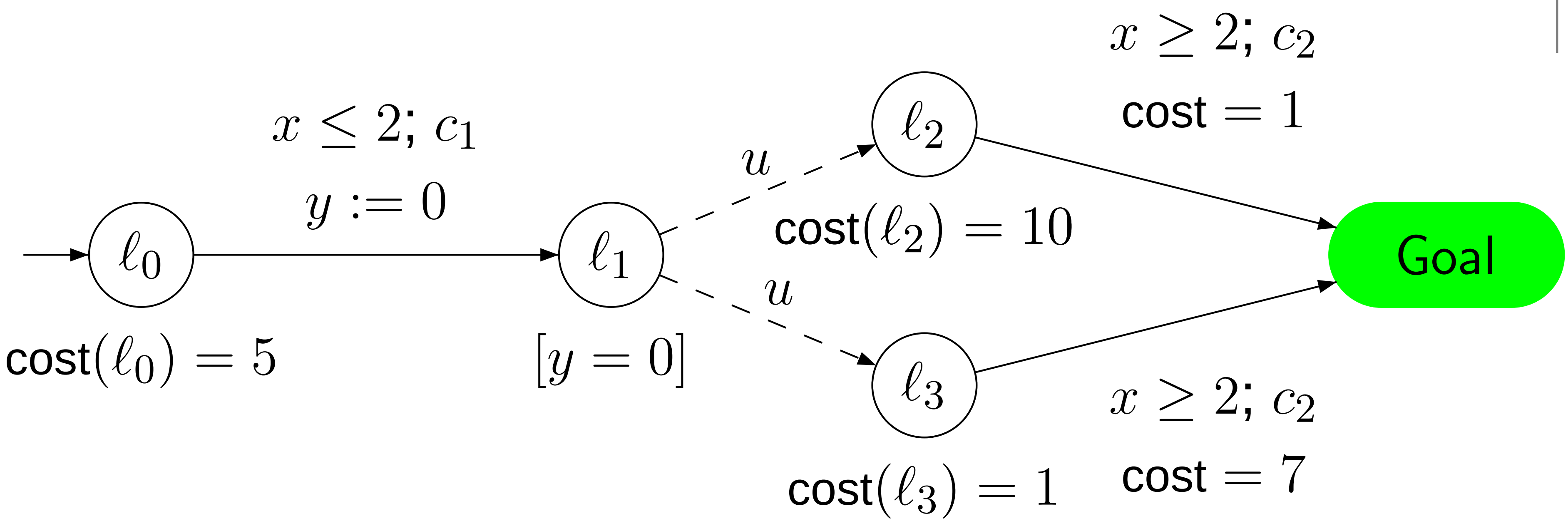
- define f_ε with $0 < \varepsilon < 1$ by:
  in ℓ_0: $f(\ell_0, x < 1 - \varepsilon) = \lambda$, $f(\ell_0, 1 - \varepsilon \leq x < 1) = c$
  in ℓ_1: $f(\ell_1, x \leq 1) = c$
  $\mathsf{Cost}(f_\varepsilon) = 1 + \varepsilon$.
- there are RPTGA for which no optimal strategy exists
- In this case there is a family of strategies f_ε such that

$$|\mathsf{Cost}((\ell_0, \vec{0}), f_\varepsilon) - \mathsf{OptCost}((\ell_0, \vec{0}), G)| < \varepsilon$$

- new problem: given ε, compute such an f_ε strategy.

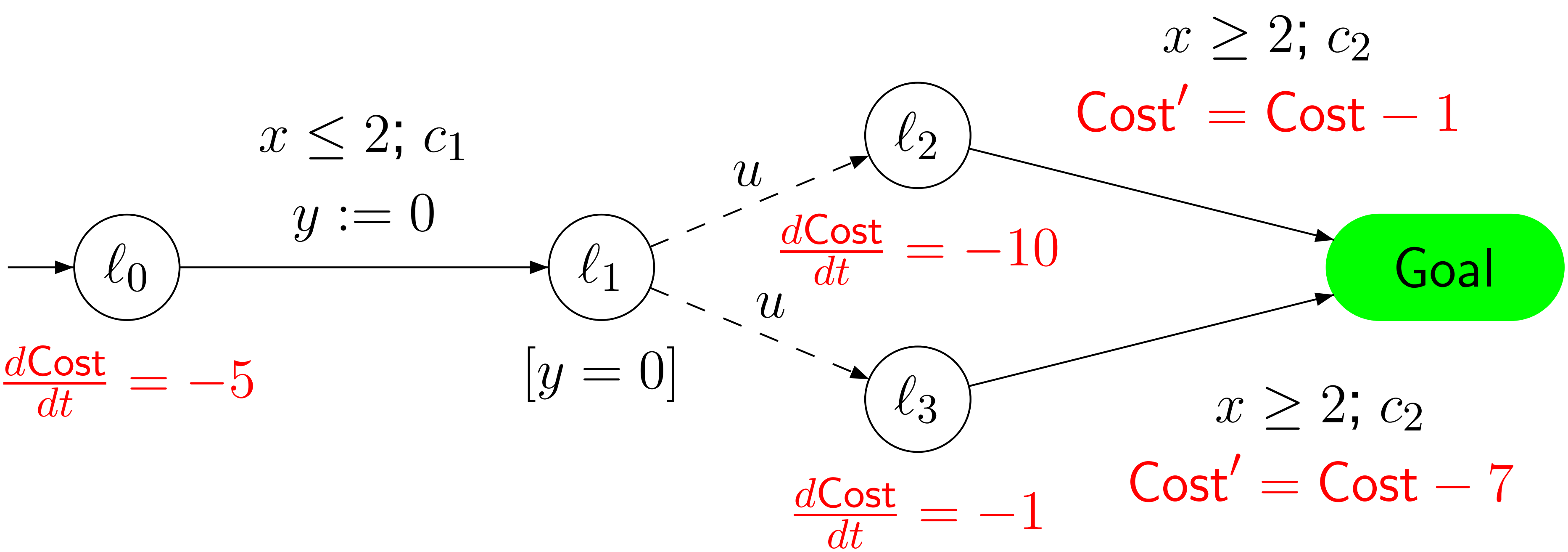
Contents

From Optimal Control to Control



A RPTGA $\mathcal{A}$

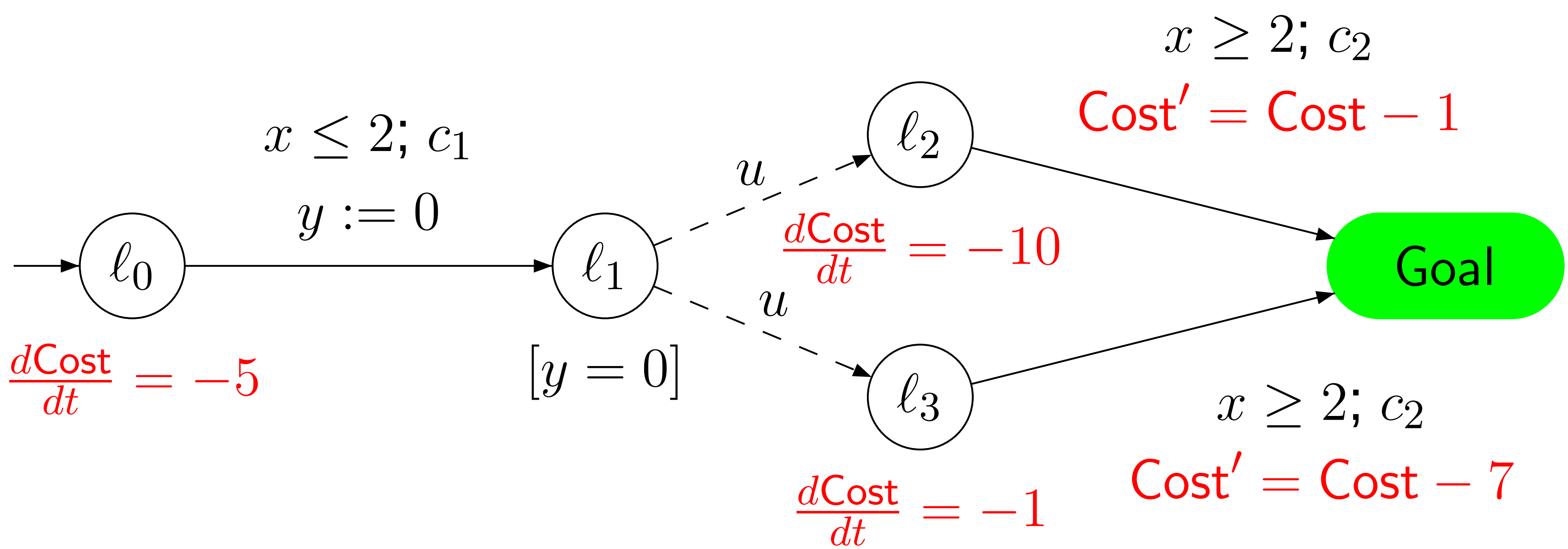
From Optimal Control to Control



- A Linear Hybrid Game Automaton $\mathcal{H}$
- Reachability Game for $\mathcal{H}$ with goal $= \mathsf{Goal} \wedge \mathsf{Cost} \geq 0$

Optimal Cost for RPTGA $\iff$ Reachability Control on LHA

From Optimal Control to Control



Assume $\exists$ semi-algorithm CompWin s.t. $W_H = \text{CompWin}(H)$ and $W_H =$ *largest set of winning states*

Theorem 2: If CompWin terminates for H then:

- it terminates for A and $W_A \stackrel{\text{def}}{=} \text{CompWin}(A) = \exists \text{Cost}.W_H$
- $(q, c) \in W_H \iff \exists f \in \text{WinStrat}(q, W_A)$ with $\text{Cost}(q, f) \leq c$

Known Results for Reachability Games

- Controllable Predecessors [MPS95, DAHM01]

$$\pi(X) = \mathsf{Pred}_t\left(X \cup \mathsf{cPred}(X), \mathsf{uPred}(\overline{X})\right)$$

[$\Longrightarrow$ Formal Def. of π]

- W (largest) set of winning states, goal $= X_0$

$$W = \mu X . X_0 \cup \pi(X)$$

Known Results for Reachability Games

- Controllable Predecessors [MPS95, DAHM01]

$$\pi(X) = \mathsf{Pred}_t\left(X \cup \mathsf{cPred}(X), \mathsf{uPred}(\overline{X})\right)$$

[$\Longrightarrow$ Formal Def. of π]

- W (largest) set of winning states, goal $= X_0$

$$W = \mu X.X_0 \cup \pi(X)$$

- π preserves Cost upward-closed sets

$$\pi(R \wedge \mathsf{Cost} \succ h) = R' \wedge \mathsf{Cost} \succ' h'$$

- semi-algorithm CompWin (preserves upwards closure)
- result of CompWin of the form $\cup_{n \in \mathbb{N}}((\ell_n, R_n \wedge \mathsf{Cost} \succ_n h_n))$ where h_n is a piece-wise affine function

Contents

Computing the Optimal Cost for PTGA

1. $\exists$ semi-algorithm $\mathsf{CompWin}$ for LHG
2. $W = \mathsf{CompWin}(H, \mathsf{Goal} \wedge \mathsf{Cost} \geq 0)$
3. $W_0 = W \cap (\ell_0, \vec{0})$
4. projection on Cost: $\exists(All \setminus \{\mathsf{Cost}\}).W_0$
   - if $\mathsf{Cost} \geq k$, $\mathsf{OptCost} = k$ and $\exists$ an optimal strategy
   - if $\mathsf{Cost} > k$, $\mathsf{OptCost} = k$ and $\exists$ a family of sub-optimal strategies

Semi-algorithm for Priced Timed Hybrid Automata

Termination ???

Termination for RPTGA

- A a RPTGA s.t. non-zeno cost: $\exists \kappa$ s.t. every cycle in the region automaton has cost at least κ
- A is bounded *i.e.* all clocks in A are bounded

Theorem 4 CompWin terminates for H, where H is the LHG associated with A [$\Longrightarrow$ Sketch of the Proof]

Termination for RPTGA

- A a RPTGA s.t. non-zeno cost: $\exists \kappa$ s.t. every cycle in the region automaton has cost at least κ
- A is bounded *i.e.* all clocks in A are bounded

Theorem 4 CompWin terminates for H, where H is the LHG associated with A [$\Longrightarrow$ Sketch of the Proof]

- Non zeno cost really needed ?
- Complexity ???

Contents

Optimal Strategy Synthesis

- $\mathcal{S}$ algorithm for synthetizing strategies for reachability timed games ? see [BCFL04] . . .
- use $\mathcal{S}$ on the LHG H: strategies are cost-dependent

Theorem 5 If $\mathcal{S}$ terminates and $\exists$ an optimal strategy we can compute a witness (cost-dependent)

Optimal Strategy Synthesis

- $\mathcal{S}$ algorithm for synthetizing strategies for reachability timed games ? see [BCFL04] ...
- use $\mathcal{S}$ on the LHG H: strategies are cost-dependent

Theorem 5 If $\mathcal{S}$ terminates and $\exists$ an optimal strategy we can compute a witness (cost-dependent)

- assume a RPTGA A is bounded, non zeno cost
- W is the set of winning states in the LHG H
- $W = \cup_{n \in \mathbb{N}}((\ell_n, R_n \wedge \mathsf{Cost} \geq h_n))$ (h_n piece-wise lin. aff.)

Theorem 6 [State-based Strategies] Let $W_A = \mathsf{CompWin}(A)$.

$\exists f$ state-based s.t. $\forall (\ell, v) \in W_A\ \mathsf{Cost}((\ell, v), f) = \mathsf{OptCost}(\ell, v)$

Synthesis of Cost-Dependent Strategies

- for LHG winning states $=$ fixed point of π operator
- $W_0 =$ Goal and $W_{i+1} = \mathsf{Pred}_t\left(W_i \cup \mathsf{cPred}(W_i), \mathsf{uPred}(\overline{W_i})\right)$

Synthesis of Cost-Dependent Strategies

- for LHG winning states $=$ fixed point of π operator
- $W_0 = \mathsf{Goal}$ and $W_{i+1} = \mathsf{Pred}_t\left(W_i \cup \mathsf{cPred}(W_i), \mathsf{uPred}(\overline{W_i})\right)$
- synthesis of cost-dependent (state-based on LHG) strategy:
  - assume f_i is a winning, state-based strategy on W_i
  - compute $W_{i+1} = \pi(W_i)$ and let $Y = W_{i+1} \setminus W_i$

Synthesis of Cost-Dependent Strategies

- for LHG winning states $=$ fixed point of π operator
- $W_0 = \mathsf{Goal}$ and $W_{i+1} = \mathsf{Pred}_t\left(W_i \cup \mathsf{cPred}(W_i), \mathsf{uPred}(\overline{W_i})\right)$
- synthesis of cost-dependent (state-based on LHG) strategy:
  - assume f_i is a winning, state-based strategy on W_i
  - compute $W_{i+1} = \pi(W_i)$ and let $Y = W_{i+1} \setminus W_i$
  - on W_i define $f_{i+1} = f_i$

Synthesis of Cost-Dependent Strategies

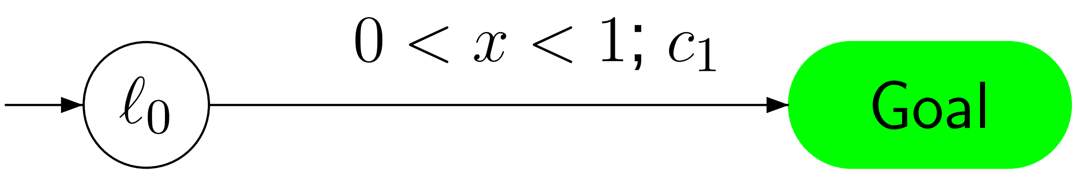
- for LHG winning states $=$ fixed point of π operator
- $W_0 = \mathsf{Goal}$ and $W_{i+1} = \mathsf{Pred}_t\big(W_i \cup \mathsf{cPred}(W_i), \mathsf{uPred}(\overline{W_i})\big)$
- synthesis of cost-dependent (state-based on LHG) strategy:
  - assume f_i is a winning, state-based strategy on W_i
  - compute $W_{i+1} = \pi(W_i)$ and let $Y = W_{i+1} \setminus W_i$
  - on W_i define $f_{i+1} = f_i$
  - on $Y_c = \mathsf{cPred}(W_i) \cap Y$ define $f_{i+1} = \{$*some* c *action*$\}$

Synthesis of Cost-Dependent Strategies

- for LHG winning states $=$ fixed point of π operator
- $W_0 = \mathsf{Goal}$ and $W_{i+1} = \mathsf{Pred}_t\left(W_i \cup \mathsf{cPred}(W_i), \mathsf{uPred}(\overline{W_i})\right)$
- synthesis of cost-dependent (state-based on LHG) strategy:
  - assume f_i is a winning, state-based strategy on W_i
  - compute $W_{i+1} = \pi(W_i)$ and let $Y = W_{i+1} \setminus W_i$
  - on W_i define $f_{i+1} = f_i$
  - on $Y_c = \mathsf{cPred}(W_i) \cap Y$ define $f_{i+1} = \{$*some* c *action*$\}$
  - on $Y_t = Y \setminus Y_c$ define $f_{i+1} = \{\lambda\}$

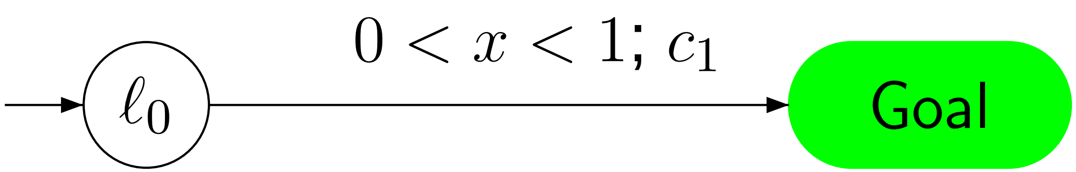
Synthesis of Cost-Dependent Strategies

- synthesis of cost-dependent (state-based on LHG) strategy:
  - assume f_i is a winning, state-based strategy on W_i
  - compute $W_{i+1} = \pi(W_i)$ and let $Y = W_{i+1} \setminus W_i$
  - on W_i define $f_{i+1} = f_i$
  - on $Y_c = \text{cPred}(W_i) \cap Y$ define $f_{i+1} = \{\textit{some } c \textit{ action}\}$
  - on $Y_t = Y \setminus Y_c$ define $f_{i+1} = \{\lambda\}$
- Problem ?



Synthesis of Cost-Dependent Strategies

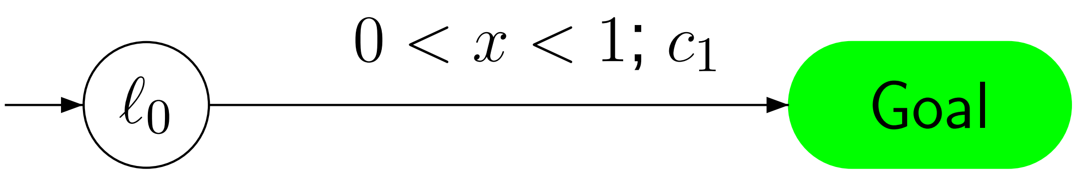
- synthesis of cost-dependent (state-based on LHG) strategy:
  - assume f_i is a winning, state-based strategy on W_i
  - compute $W_{i+1} = \pi(W_i)$ and let $Y = W_{i+1} \setminus W_i$
  - on W_i define $f_{i+1} = f_i$
  - on $Y_c = \text{cPred}(W_i) \cap Y$ define $f_{i+1} = \{$*some* c *action*$\}$
  - on $Y_t = Y \setminus Y_c$ define $f_{i+1} = \{\lambda\}$
- Problem ?



  - $W_1 = \{\text{Goal}\} \cup \{(\ell_0, 0 \leq x < 1)\}$ and $Y = (\ell_0, 0 \leq x < 1)$
  - $f_1(\ell_0, 0 < x < 1) = \{c_1\}$ and $f_1(\ell_0, x = 0) = \{\lambda\}$
  - blocking strategy

Synthesis of Cost-Dependent Strategies

- synthesis of cost-dependent (state-based on LHG) strategy:
  - assume f_i is a winning, state-based strategy on W_i
  - compute $W_{i+1} = \pi(W_i)$ and let $Y = W_{i+1} \setminus W_i$
  - on W_i define $f_{i+1} = f_i$
  - on $Y_c = \text{cPred}(W_i) \cap Y$ define $f_{i+1} = \{\textit{some } c \textit{ action}\}$
  - on $Y_t = Y \setminus Y_c$ define $f_{i+1} = \{\lambda\}$
- Problem ?



  - Choose $\varepsilon > 0$
  - $f_1(\ell_0, \varepsilon \leq x < 1) = \{c_1\}$ and $f_1(\ell_0, 0 \leq x < \varepsilon) = \{\lambda\}$
  - new winning, state-based strategy

Synthesis of Cost-Dependent Strategies

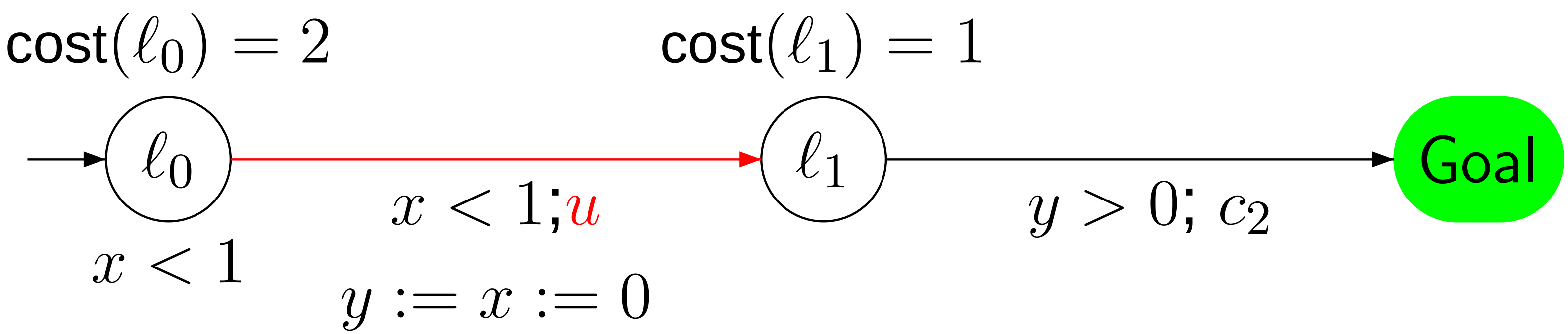
- synthesis of cost-dependent (state-based on LHG) strategy:
  - assume f_i is a winning, state-based strategy on W_i
  - compute $W_{i+1} = \pi(W_i)$ and let $Y = W_{i+1} \setminus W_i$
  - on W_i define $f_{i+1} = f_i$
  - on $Y_c = \text{cPred}(W_i) \cap Y$ define $f_{i+1} = \{\textit{some } c \textit{ action}\}$
  - on $Y_t = Y \setminus Y_c$ define $f_{i+1} = \{\lambda\}$
- Computation of a winning state-based strategy:
  - if guards of u actions are strict and guards on c actions are large then f_{i+1} is winning (Y_t is future-open)
  - otherwise f_{i+1} can be altered to be made winning
  - consequence: if $\pi^*(W_0) = W_k$ for some $k \in \mathbb{N}$ there is a winning state-based (cost-dependent) strategy

Optimal Cost-Independent Strategy

- compute a cost-dependent winning strategy f;
  $f(q, cost) \in \mathsf{Act}_c \cup \{\lambda\}$
- Optimal cost-independent winning strategy f^*:
  - take the best action in each state: $f^*(q) = e$ if
    1. $e = f(q, cost)$
    2. $\forall e' \neq e, f(q, cost') = e' \implies cost' \geq cost$
- result: under strictness assumptions, we can build a uniform optimal strategy i.e. optimal in each state (non blocking)

[$\Longrightarrow$ Algorithm & HyTech]

No Optimal Cost-Independent Strategy

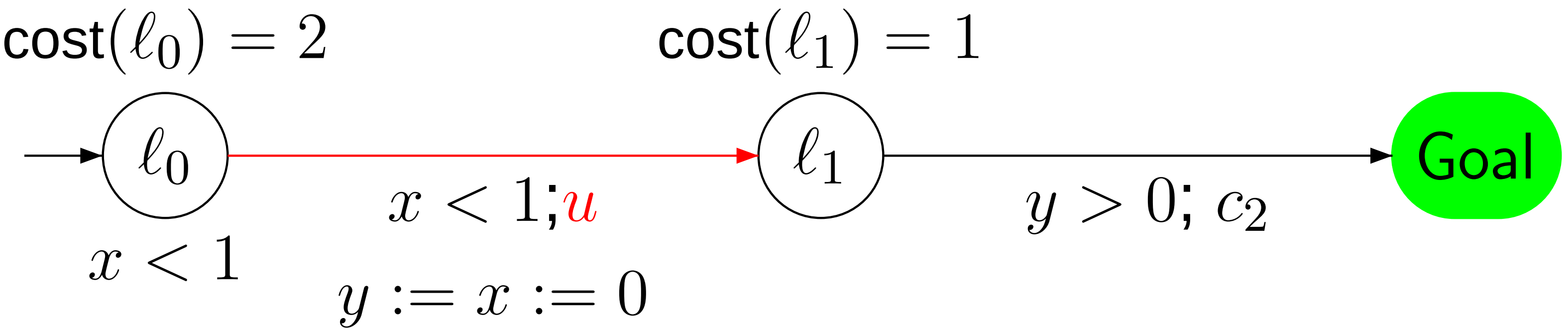


- Optimal cost is 2
- An optimal winning cost-dependent strategy f:
  $f(\ell_1, -, cost < 2) = \lambda$ and $f(\ell_1, -, cost = 2) = c_2$
  assume u taken at time $(1 - \delta_0)$:

$$\text{Cost}(f, (\ell_0, 0)) = 2 \cdot (1 - \delta_0) + \delta_1$$

and according to f we have $\delta_1 = 2 \cdot \delta_2$

No Optimal Cost-Independent Strategy



- Optimal cost is 2
- assume $\exists\ f^*$ cost-independent: f^* must wait in ℓ_1 at least ε
  assume u taken at time $(1 - \delta)$:

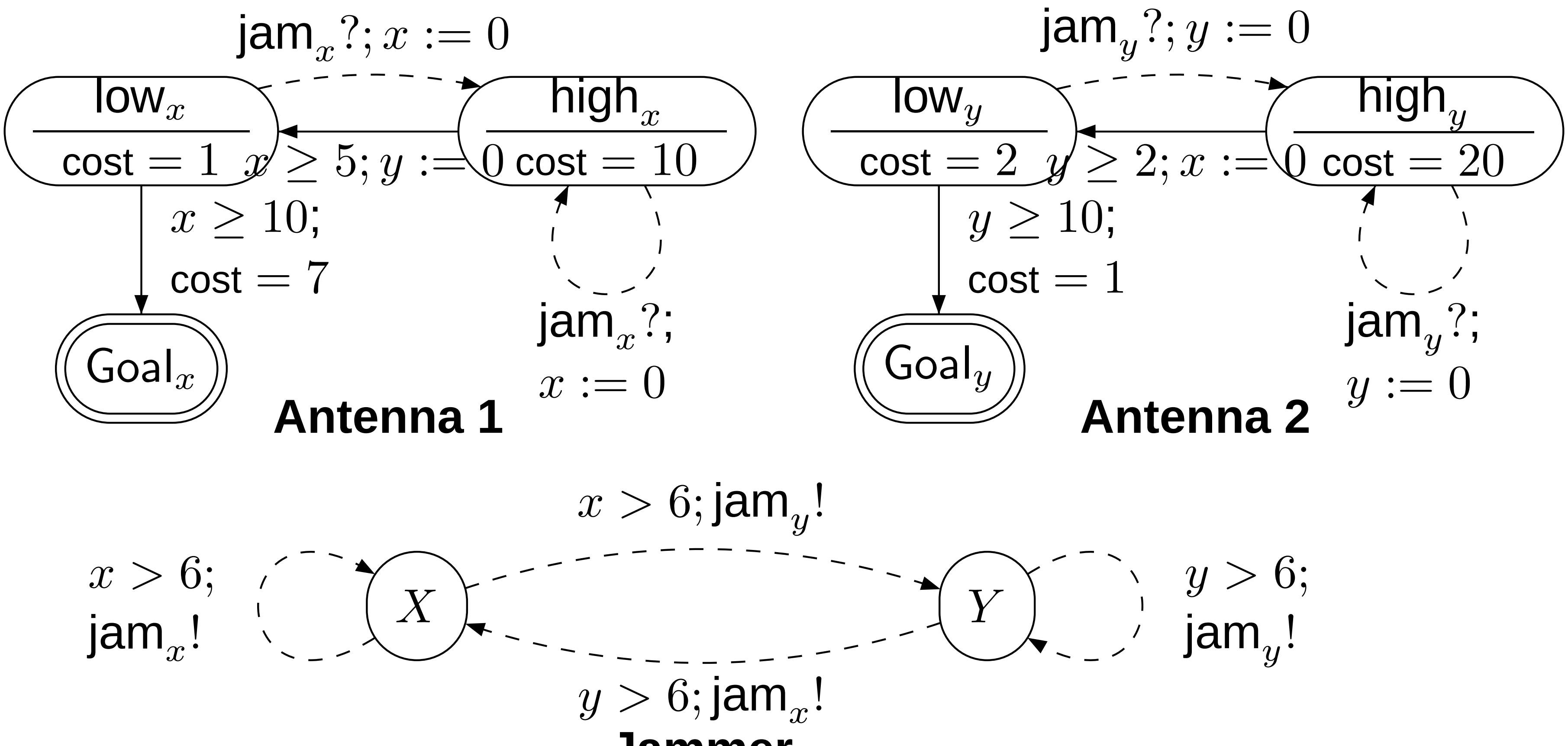
$$\text{Cost}(f^*, (\ell_0, 0)) = 2 \cdot (1 - \delta) + \varepsilon$$

Take $\delta = \frac{\varepsilon}{4}$: $\text{Cost}(f^*, (\ell_0, 0)) = 2 + \frac{\varepsilon}{2}$ and $\text{OptCost}(f^*) = 2 + \varepsilon$

Contents

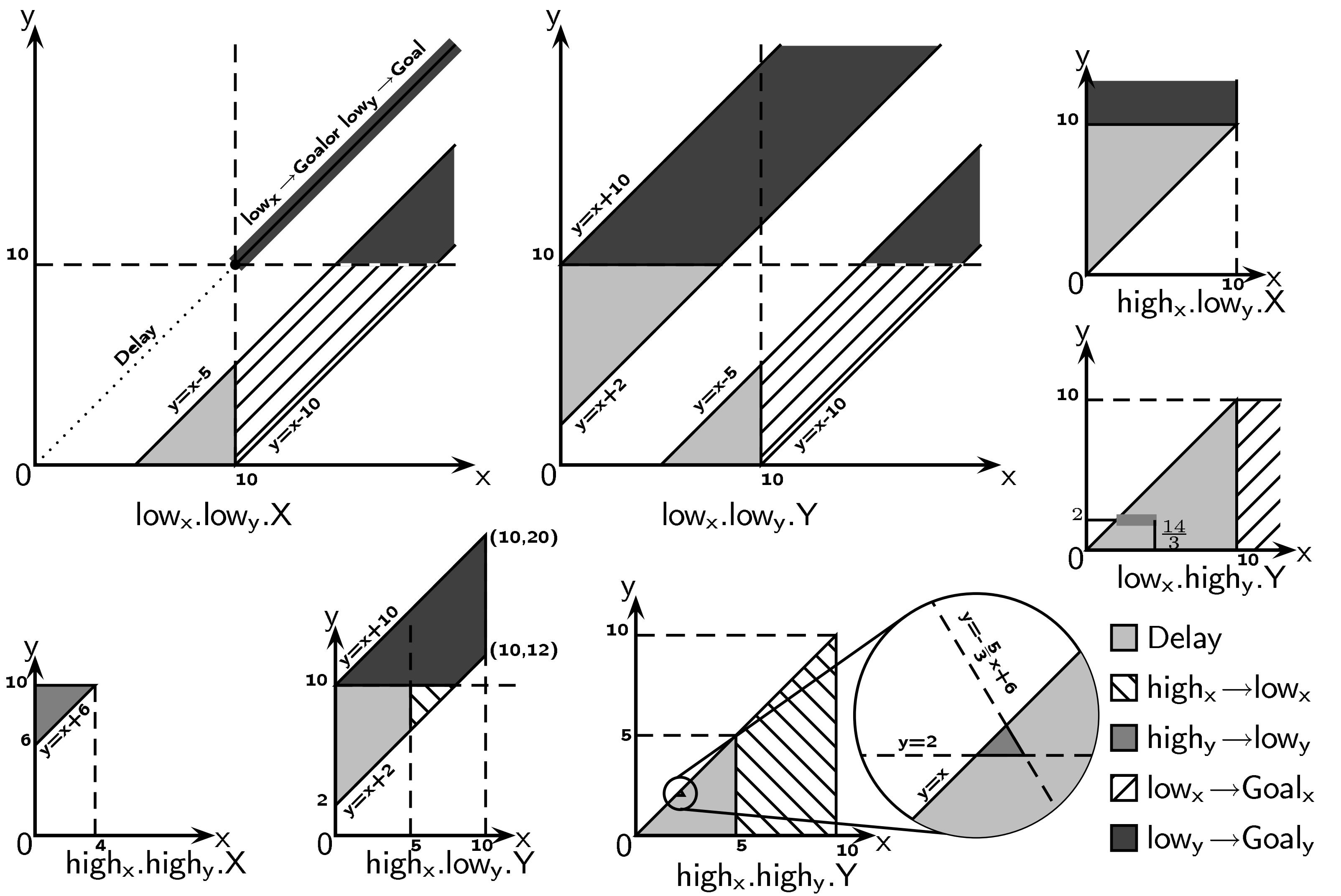
Experiment

- computation of optimal cost and optimal strategies (if ∃) implemented in HyTech (Demo ?)
- a cyclic example: [⟹ See the strategy]



Optimal Strategy for the Mobile Phone

Optimal cost is 109



Conclusion & Future Work

Current State of Our Work

- Semi-algorithm for computing the optimal cost for LHG
- in case it terminates:
  - decide if $\exists$ optimal strategy
  - compute an optimal strategy
- Implementation in HyTech

Open Problems

- Optimal Control – Decidability issues (non zeno cost)
- maximal class for which CompWin terminates

Future Work

- compute f_ε strategies
- safety games . . .

References (1)

[AD94] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science (TCS)*, 126(2):183–235, 1994. 3

[ALTP01] R. Alur, S. La Torre, and G. J. Pappas. Optimal paths in weighted timed automata. In *Proc. 4th Int. Work. Hybrid Systems: Computation and Control (HSCC'01)*, volume 2034 of *LNCS*, pages 49–62. Springer, 2001. 3

[ABM04] R. Alur, M, Bernadsky, and P. Madhusudan. Optimal reachability in weighted timed games. In *Proc. 31st International Colloquium on Automata, Languages and Programming (ICALP'04)*, Lecture Notes in Computer Science. Springer, 2004. To appear. 5

[AM99] E. Asarin and O. Maler. As soon as possible: Time optimal control for timed automata. In *Proc. 2nd Int. Work. Hybrid Systems: Computation and Control (HSCC'99)*, volume 1569 of *LNCS*, pages 19–30. Springer, 1999. 3

[AMPS98] E. Asarin, O. Maler, A. Pnueli, and J. Sifakis. Controller synthesis for timed automata. In *Proc. IFAC Symposium on System Structure and Control*, pages 469–474. Elsevier Science, 1998. 3

[BCFL04] P. Bouyer, F. Cassez, E. Fleury, and K. G. Larsen. Optimal strategies in priced timed game automata. BRICS Report Series, Basic Research In Computer Science, Denmark, February 2004. 5, 19

References (2)

[DAHM01] L. De Alfaro, T. A. Henzinger, and R. Majumdar. Symbolic algorithms for infinite-state games. In *Proc. 12th International Conference on Concurrency Theory (CONCUR'01)*, volume 2154 of *LNCS*, pages 536–550. Springer, 2001. 14

[LBB+01] K. G. Larsen, G. Behrmann, E. Brinksma, A. Fehnker, T. Hune, P. Pettersson, and J. Romijn. As cheap as possible: Efficient cost-optimal reachability for priced timed automata. In *Proc. 13th International Conference on Computer Aided Verification (CAV'01)*, volume 2102 of *LNCS*, pages 493–505. Springer, 2001. 3

[LTMM02] S. La Torre, S. Mukhopadhyay, and A. Murano. Optimal-reachability and control for acyclic weighted timed automata. In *Proc. 2nd IFIP International Conference on Theoretical Computer Science (TCS 2002)*, volume 223 of *IFIP Conference Proceedings*, pages 485–497. Kluwer, 2002. 5, 10

[MPS95] O. Maler, A. Pnueli, and J. Sifakis. On the synthesis of discrete controllers for timed systems. In *Proc. 12th Annual Symposium on Theoretical Aspects of Computer Science (STACS'95)*, volume 900, pages 229–242. Springer, 1995. 3, 14

Recursive Definition of Optimal Cost

Let G be a RPTG. Let O be the function from Q to $\mathbb{R}_{\geq 0} \cup \{+\infty\}$ that is the least fixed point of the following functional:

$$O(q)? \quad q \xrightarrow{t,p} q' \quad \max \left\{ \vphantom{\begin{array}{c}a\\b\\c\\d\\e\\f\\g\\h\end{array}} \right.$$

Recursive Definition of Optimal Cost

Let G be a RPTG. Let O be the function from Q to $\mathbb{R}_{\geq 0} \cup \{+\infty\}$ that is the least fixed point of the following functional:

$$O(q)? \quad q \xrightarrow{t,p} q' \quad \max \left\{ \min\left(\left(\min_{\substack{q' \xrightarrow{c,p'} q'' \\ c \in \mathsf{Act}_c}} p + p' + O(q'') \right), p + O(q') \right) \right.$$

- Controllable actions in q'

Recursive Definition of Optimal Cost

Let G be a RPTG. Let O be the function from Q to $\mathbb{R}_{\geq 0} \cup \{+\infty\}$ that is the least fixed point of the following functional:

$$O(q)? \quad q \xrightarrow{t,p} q' \quad \max \begin{cases} \min\left(\left(\min_{\substack{q' \xrightarrow{c,p'} q'' \\ c \in \mathsf{Act}_c}} p + p' + O(q'')\right), p + O(q')\right) \\ \sup_{\substack{q \xrightarrow{t',p'} q'' \\ t' \leq t}} \max_{\substack{q'' \xrightarrow{u,p''} q''' \\ u \in \mathsf{Act}_u}} p' + p'' + O(q''') \end{cases}$$

- Controllable actions in q'
- Uncontrollable actions before q'

Recursive Definition of Optimal Cost

Let G be a RPTG. Let O be the function from Q to $\mathbb{R}_{\geq 0} \cup \{+\infty\}$ that is the least fixed point of the following functional:

$$O(q) = \inf_{\substack{q \xrightarrow{t,p} q' \\ t \in \mathbb{R}_{\geq 0}}} \max \begin{cases} \min\left(\left(\min_{\substack{q' \xrightarrow{c,p'} q'' \\ c \in \mathsf{Act}_c}} p + p' + O(q'') \right), p + O(q') \right) \\ \sup_{\substack{q \xrightarrow{t',p'} q'' \\ t' \leq t}} \max_{\substack{q'' \xrightarrow{u,p''} q''' \\ u \in \mathsf{Act}_u}} p' + p'' + O(q''') \end{cases}$$

- Controllable actions in q'
- Uncontrollable actions before q'
- Minimize over t

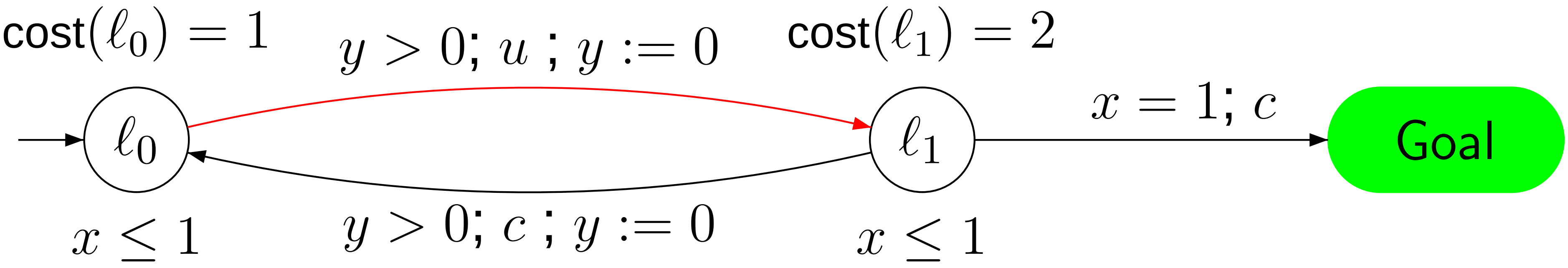
Outcome

Let $G = (L, \ell_0, \mathsf{Act}, X, E, \mathsf{inv}, \mathsf{cost})$ be a (R)PTGA and f a strategy over G. The outcome $\mathsf{Outcome}((\ell, v), f)$ of f from configuration (ℓ, v) in G is the subset of $\mathsf{Runs}((\ell, v), G)$ defined inductively by:

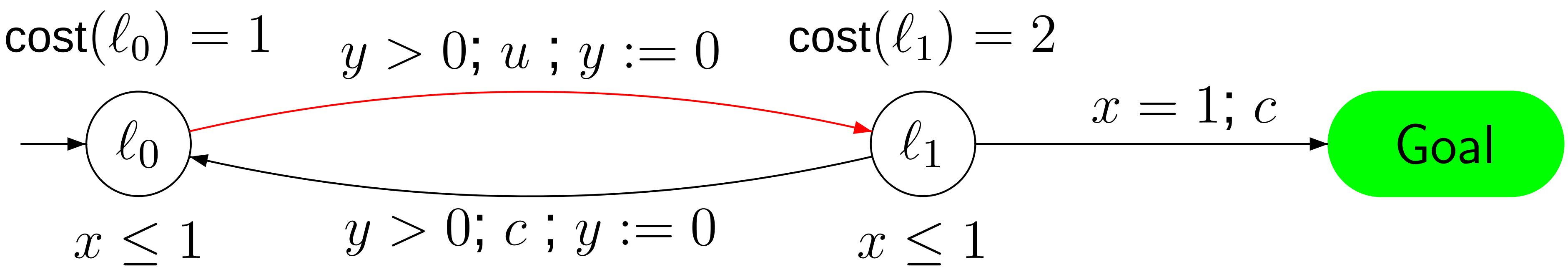
- $(\ell, v) \in \mathsf{Outcome}((\ell, v), f)$,
- if $\rho \in \mathsf{Outcome}((\ell, v), f)$ then $\rho' = \rho \xrightarrow{e} (\ell', v') \in \mathsf{Outcome}((\ell, v), f)$ if $\rho' \in \mathsf{Runs}((\ell, v), G)$ and one of the following three conditions hold:
  1. $e \in \mathsf{Act}_u$,
  2. $e \in \mathsf{Act}_c$ and $e = f(\rho)$,
  3. $e \in \mathbb{R}_{\geq 0}$ and $\forall 0 \leq e' < e, \exists (\ell'', v'') \in (L \times \mathbb{R}^X_{\geq 0})$ s.t. $last(\rho) \xrightarrow{e'} (\ell'', v'') \wedge f(\rho \xrightarrow{e'} (\ell'', v'')) = \lambda$.
- an infinite run ρ is in $\in \mathsf{Outcome}((\ell, v), f)$ if all the finite prefixes of ρ are in $\mathsf{Outcome}((\ell, v), f)$.

[⟹ Back to Strategies]

A Tricky Example

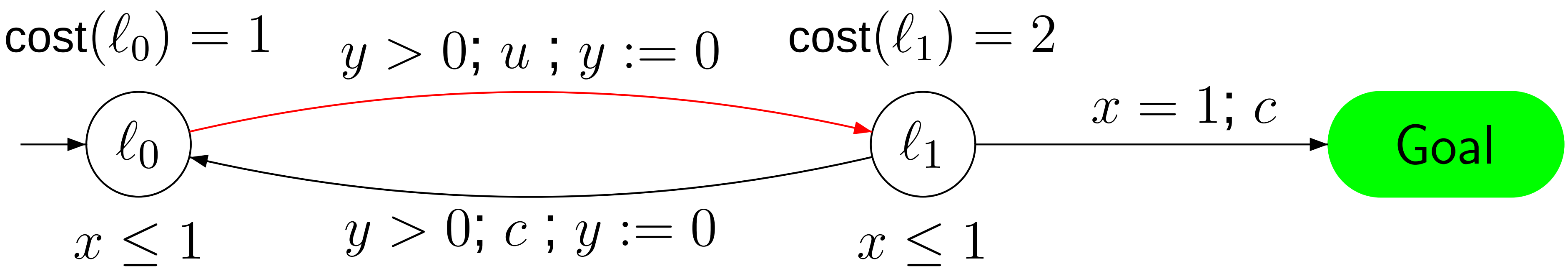


A Tricky Example



- what is the optimal cost?
- Is there an optimal strategy?

A Tricky Example



- what is the optimal cost?
- Is there an optimal strategy?
- ... assume you start with 2 ... start with less than 2 ($2 - \epsilon$)

π Operator

- (Un)Controllable Predecessors

$$\mathsf{Pred}^a(X) = \{q \in Q \mid q \xrightarrow{a} q', q' \in X\}$$

$$\mathsf{cPred}(X) = \bigcup_{c \in \mathsf{Act}_c} \mathsf{Pred}^c(X) \quad \mathsf{uPred}(X) = \bigcup_{u \in \mathsf{Act}_u} \mathsf{Pred}^u(X)$$

- Safe Time Predecessors $\mathsf{Pred}_t(X, Y)$

$$= \{q \in Q \mid \exists \delta \in \mathbb{R}_{\geq 0} \mid q \xrightarrow{\delta} q',\ q' \in X \wedge \mathsf{Post}_{[0,\delta]}(q) \subseteq \overline{Y}\}$$

$$\mathsf{Post}_{[0,\delta]}(q) = \{q' \in Q \mid \exists t \in [0, \delta] \mid q \xrightarrow{t} q'\}$$

- π Operator (uncontrollable actions "cannot win"):

$$\pi(X) = \mathsf{Pred}_t\left(X \cup \mathsf{cPred}(X), \mathsf{uPred}(\overline{X})\right)$$

π Operator

- (Un)Controllable Predecessors

$$\mathsf{Pred}^a(X) = \{q \in Q \mid q \xrightarrow{a} q', q' \ \in X\}$$

$$\mathsf{cPred}(X) = \bigcup_{c \in \mathsf{Act}_c} \mathsf{Pred}^c(X) \quad \mathsf{uPred}(X) = \bigcup_{u \in \mathsf{Act}_u} \mathsf{Pred}^u(X)$$

- Safe Time Predecessors $\mathsf{Pred}_t(X, Y)$

$$= \{q \in Q \mid \exists \delta \in \mathbb{R}_{\geq 0} \mid q \xrightarrow{\delta} q',\ q' \in X \wedge \mathsf{Post}_{[0,\delta]}(q) \subseteq \overline{Y}\}$$

$$\mathsf{Post}_{[0,\delta]}(q) = \{q' \in Q \mid \exists t \in [0, \delta] \mid q \xrightarrow{t} q'\}$$

- π': uncontrollable actions sometimes can win:

$$\pi'(X) = \pi(X) \cup \mathsf{Pred}_t\left(\mathsf{uPred}(X) \cap STOP, \mathsf{uPred}(\overline{X})\right)$$

π Operator

- (Un)Controllable Predecessors

$$\mathsf{Pred}^a(X) = \{q \in Q \mid q \xrightarrow{a} q', q' \in X\}$$

$$\mathsf{cPred}(X) = \bigcup_{c \in \mathsf{Act}_c} \mathsf{Pred}^c(X) \quad \mathsf{uPred}(X) = \bigcup_{u \in \mathsf{Act}_u} \mathsf{Pred}^u(X)$$

- Safe Time Predecessors $\mathsf{Pred}_t(X, Y)$

$$= \{q \in Q \mid \exists \delta \in \mathbb{R}_{\geq 0} \mid q \xrightarrow{\delta} q',\ q' \in X \wedge \mathsf{Post}_{[0,\delta]}(q) \subseteq \overline{Y}\}$$

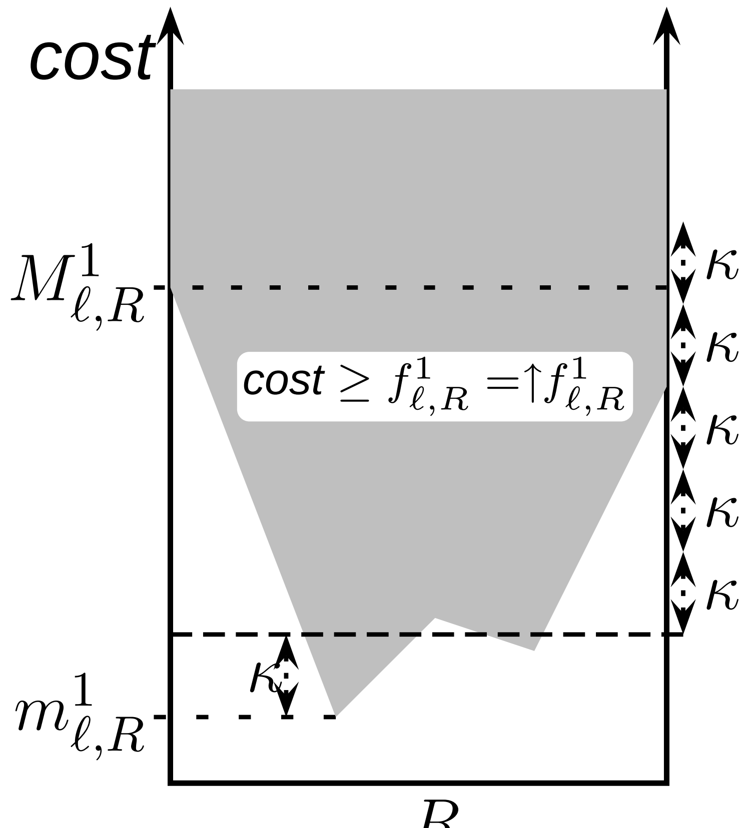
$$\mathsf{Post}_{[0,\delta]}(q) = \{q' \in Q \mid \exists t \in [0, \delta] \mid q \xrightarrow{t} q'\}$$

- π'': uncontrollable actions bound to happen win:

$$\pi''(X) = \pi(X) \cup \mathsf{Pred}_t\left(Inv \cap \overline{\mathsf{Pred}_t(\overline{\mathsf{uPred}(X)} \cap Inv)}, \mathsf{uPred}(\overline{X})\right)$$

Termination Criterion for RPTGA

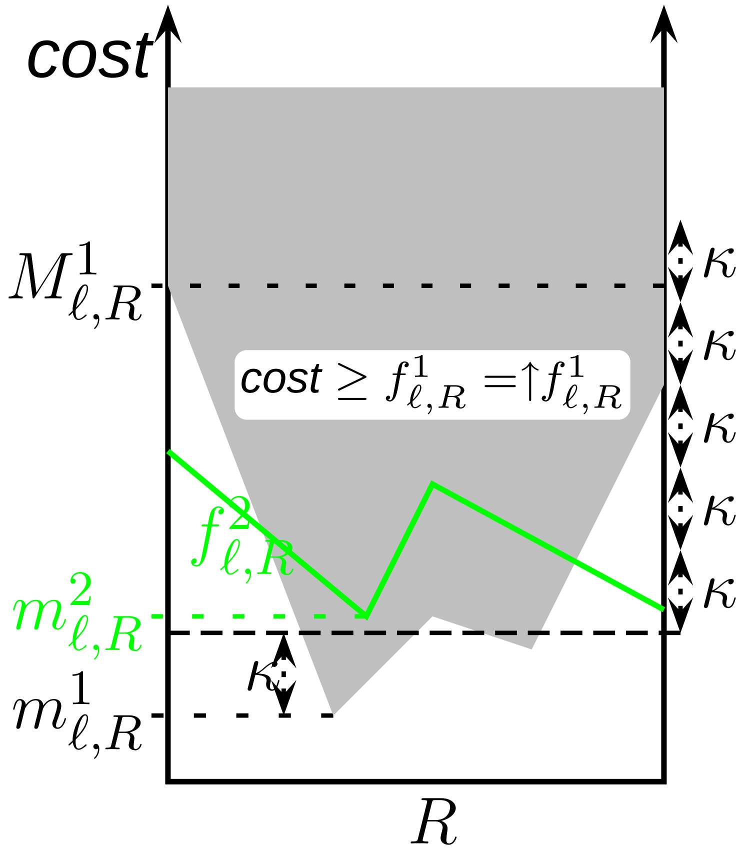
- R is a (bounded) region of the region automaton (RA)
- every cycle in the RA costs at least κ



[$\Longrightarrow$ Back to Termination]

Termination Criterion for RPTGA

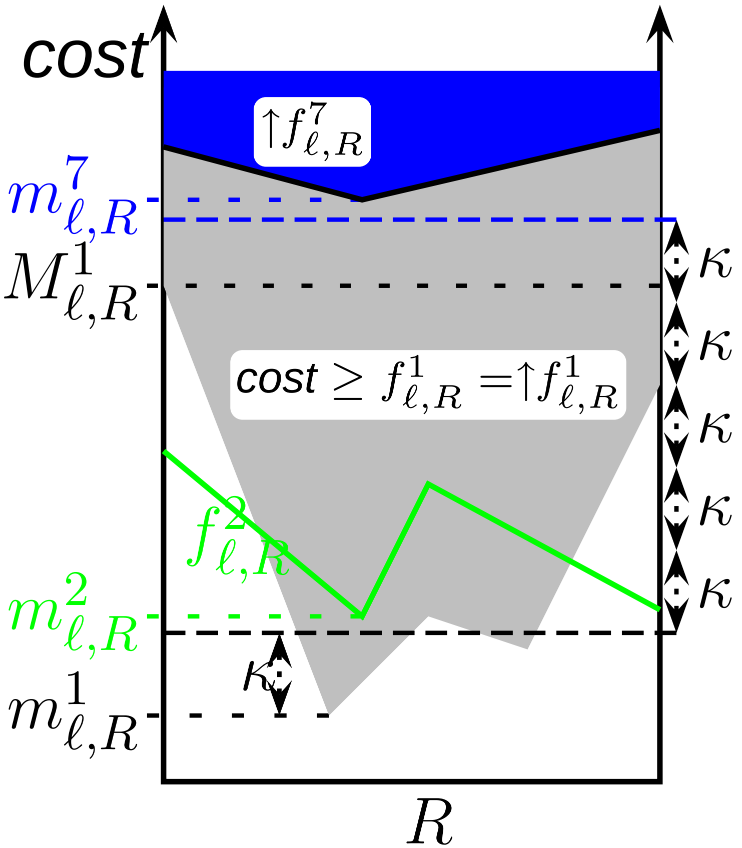
- R is a (bounded) region of the region automaton (RA)
- every cycle in the RA costs at least κ



[$\Longrightarrow$ Back to Termination]

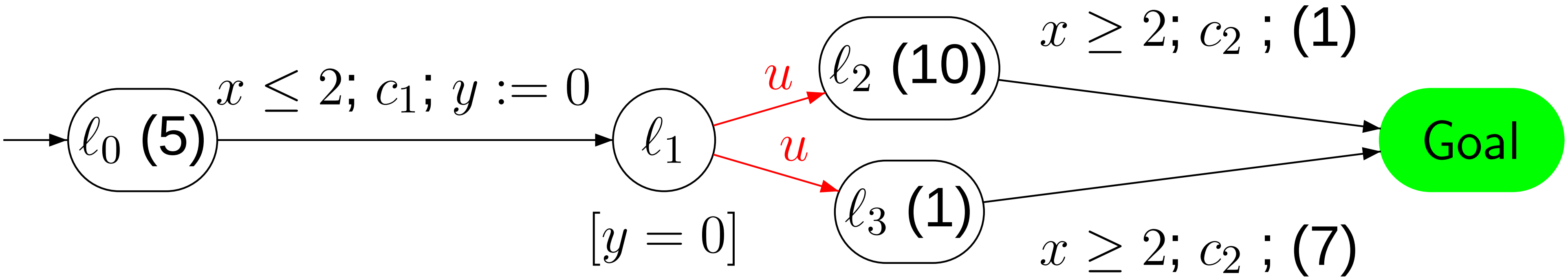
Termination Criterion for RPTGA

- R is a (bounded) region of the region automaton (RA)
- every cycle in the RA costs at least κ

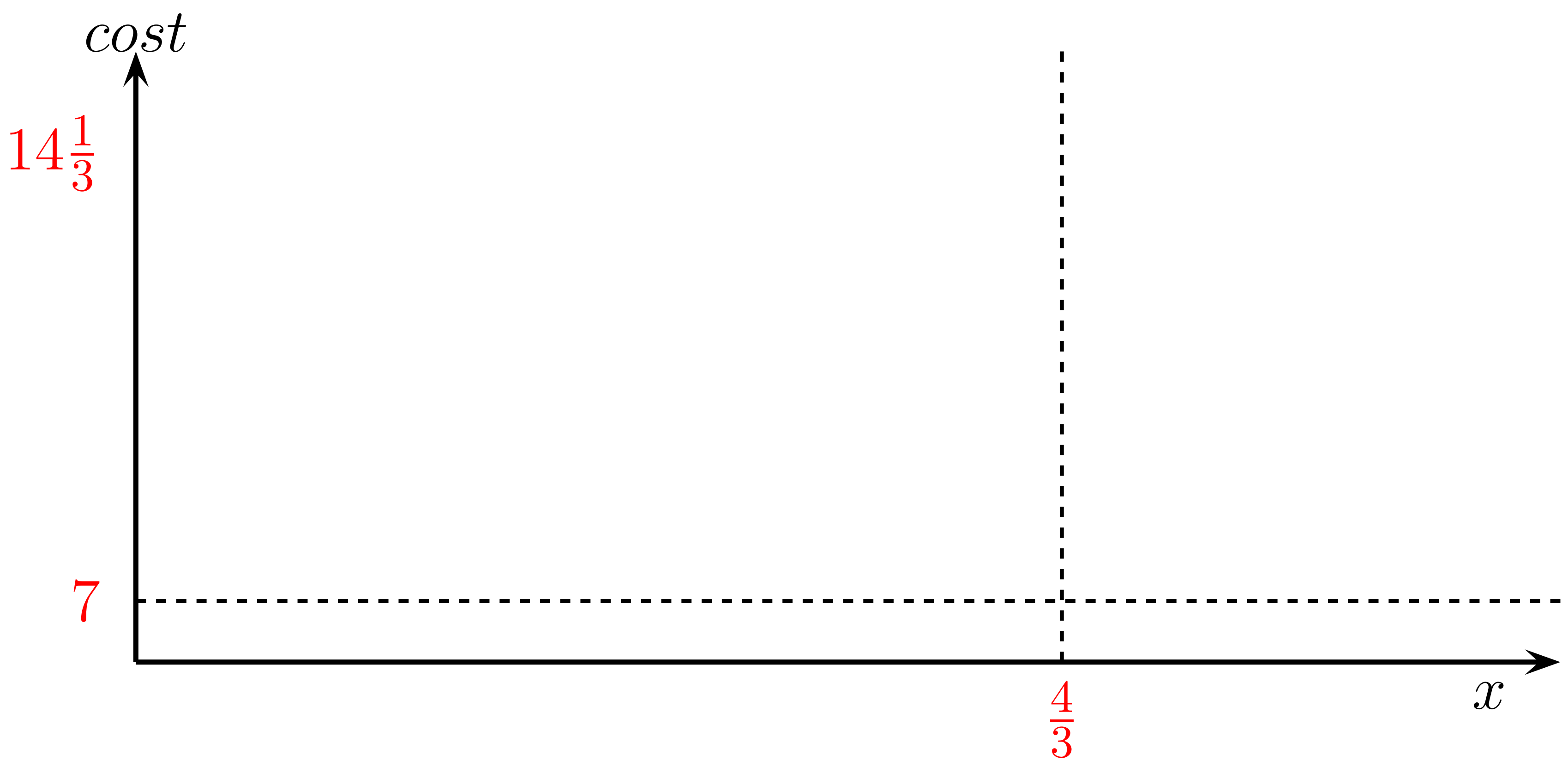


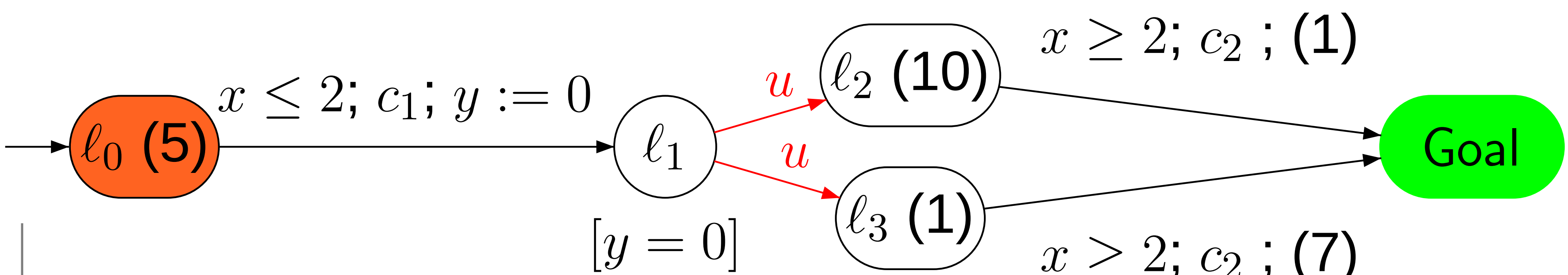
[$\Longrightarrow$ Back to Termination]

Optimal Cost-Independent Strategy

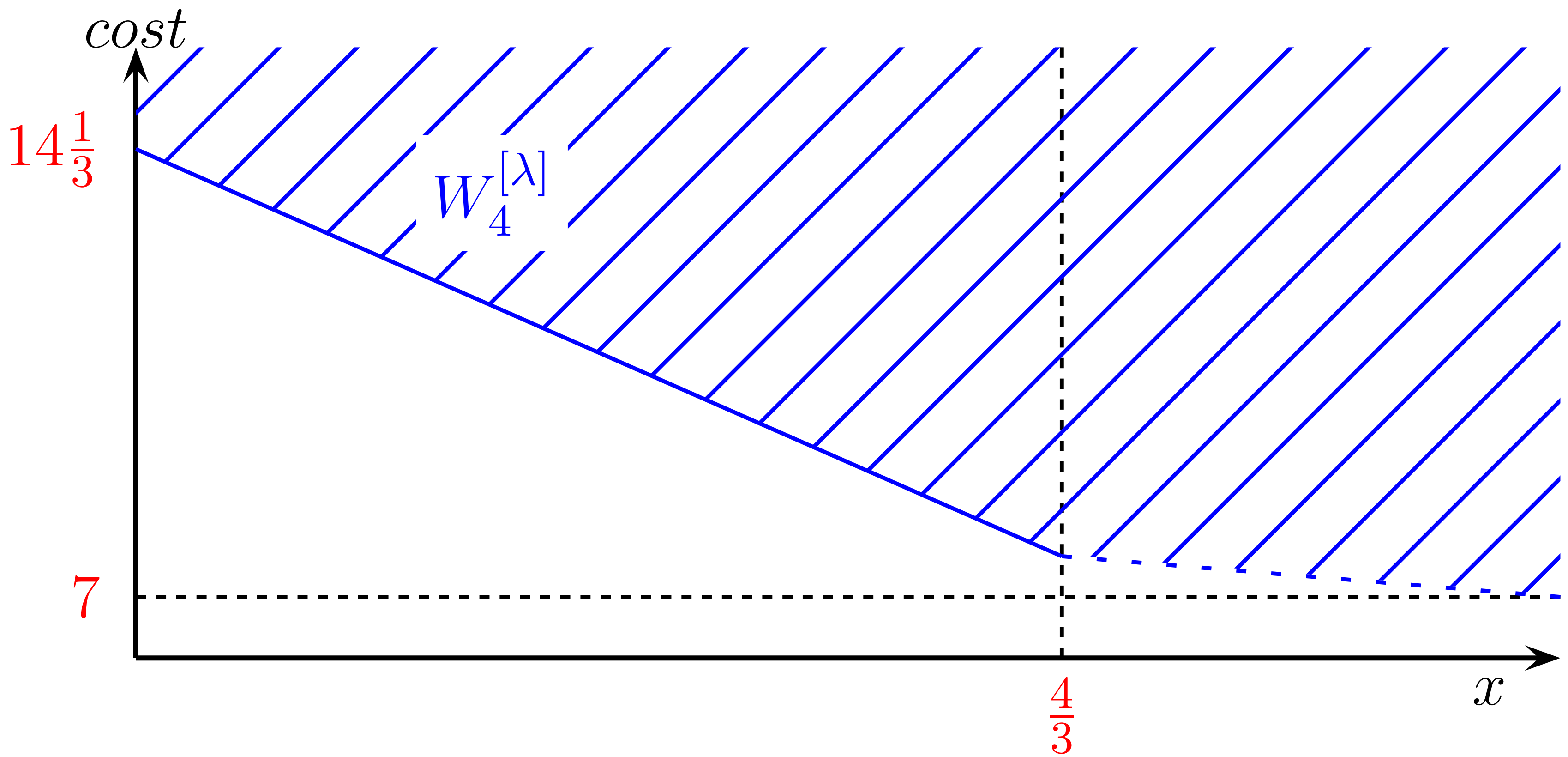


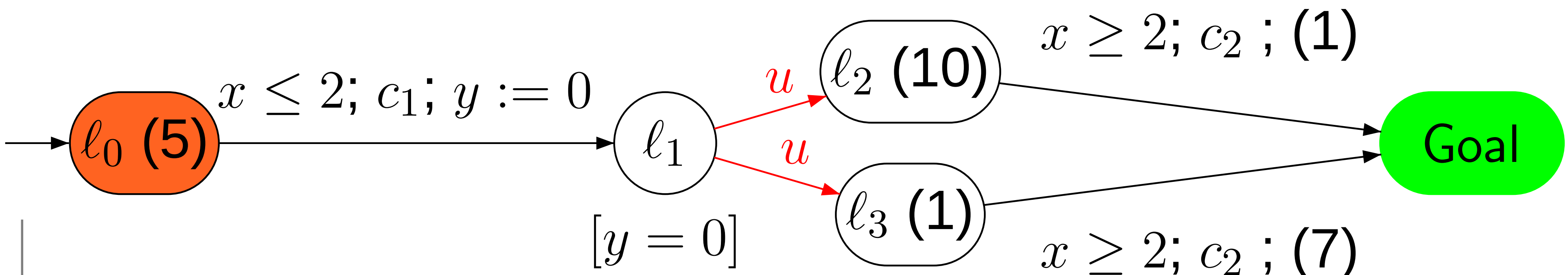
Optimal Cost-Independent Strategy



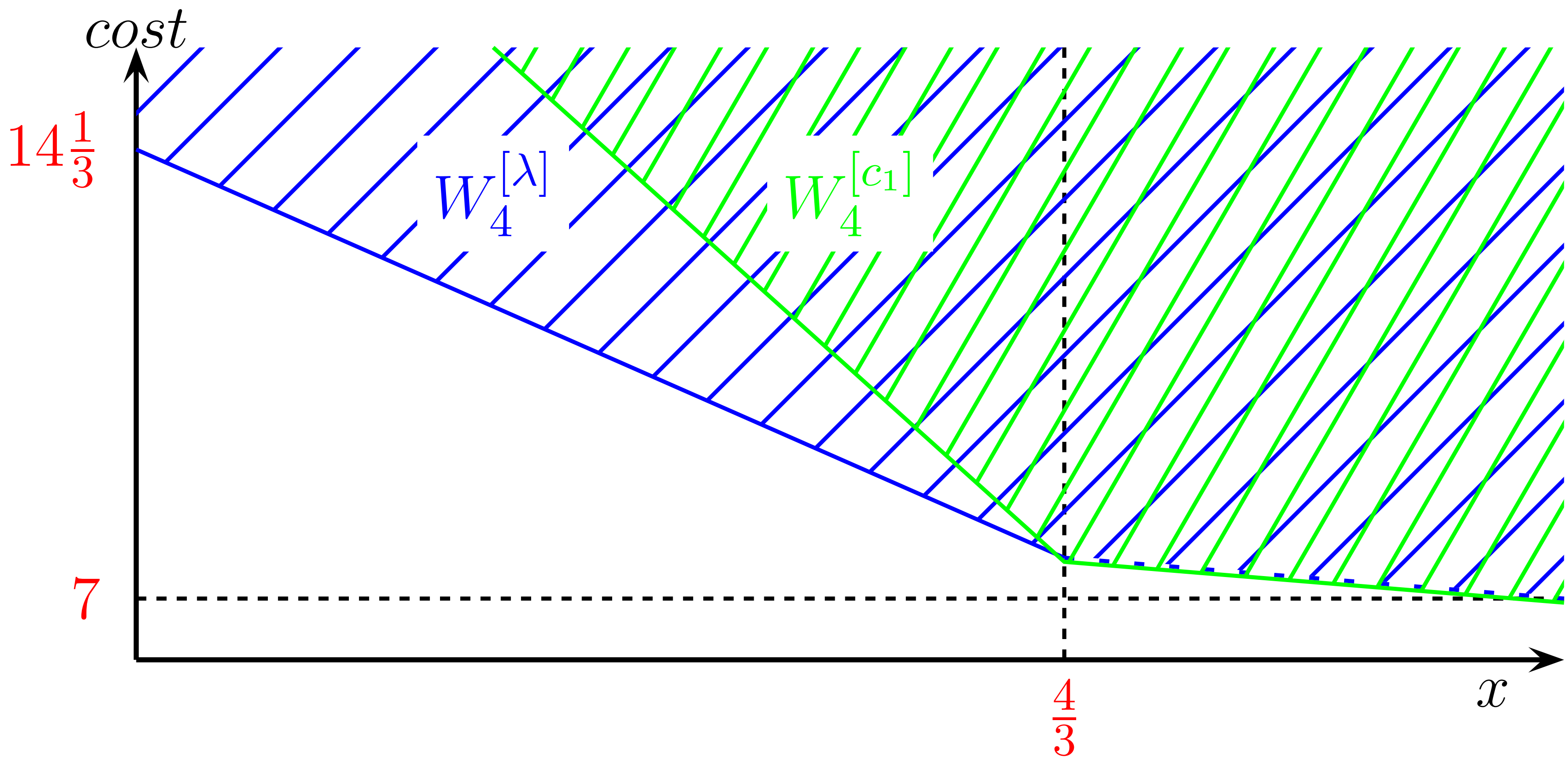


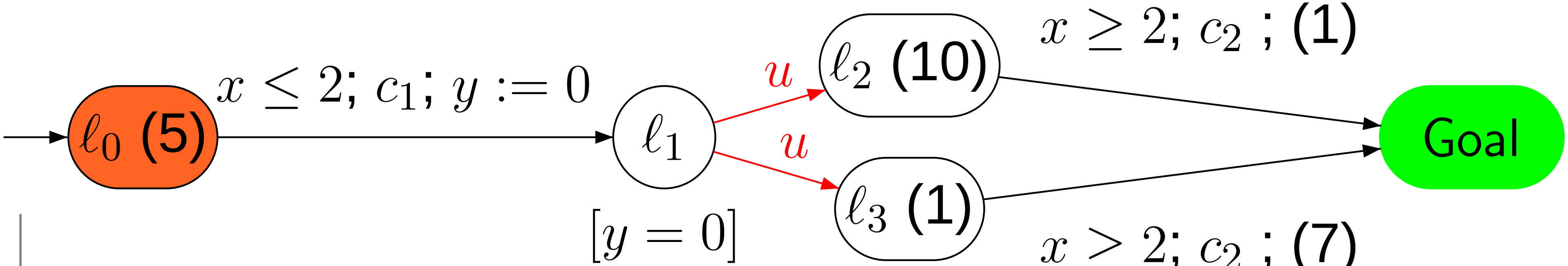
Optimal Cost-Independent Strategy



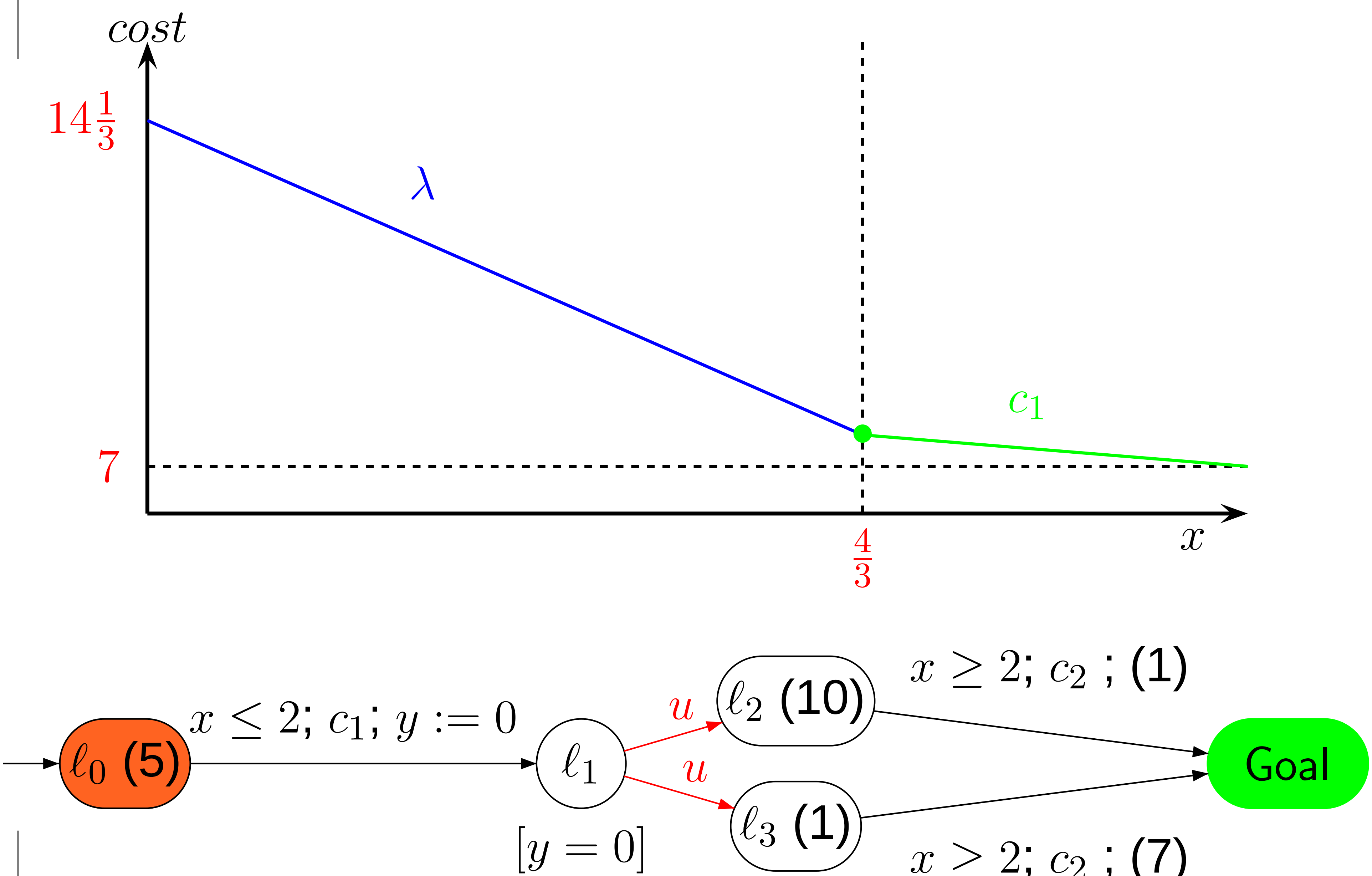


Optimal Cost-Independent Strategy





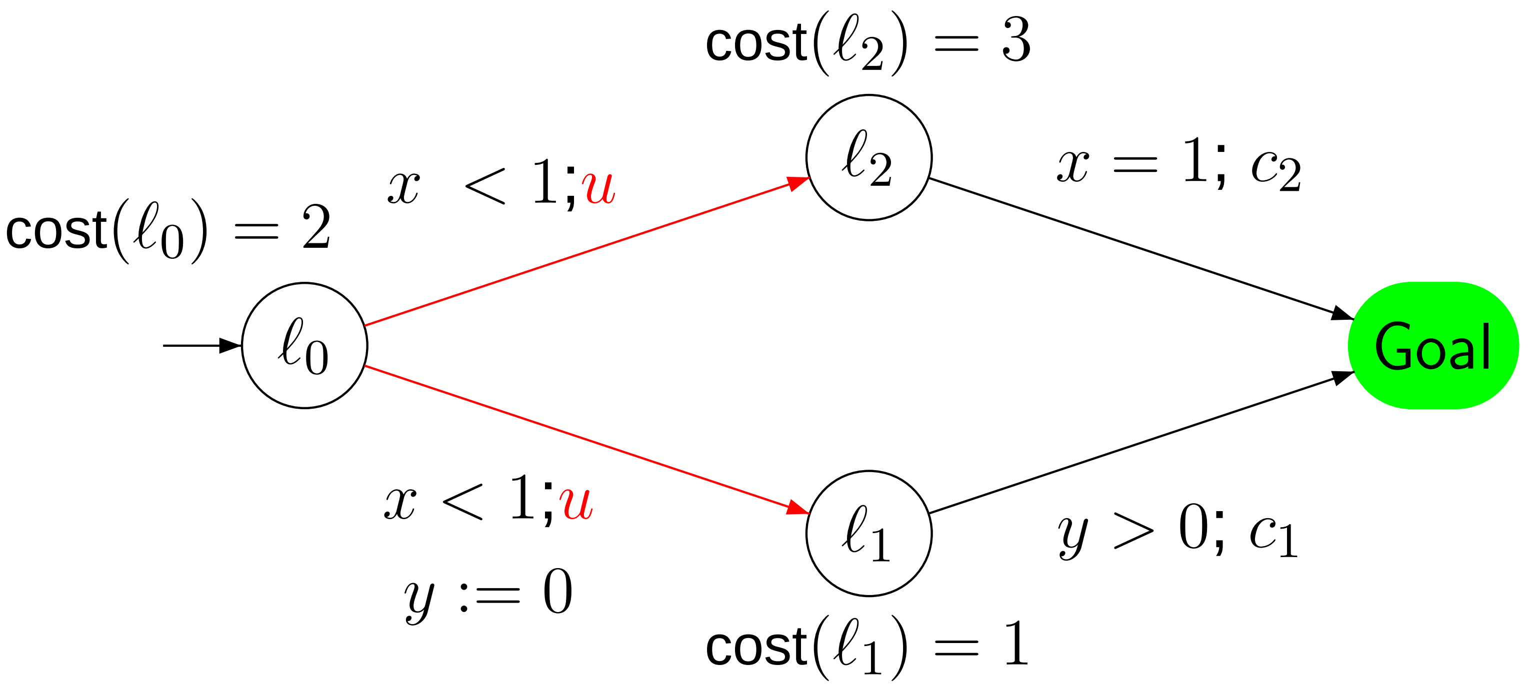
Optimal Cost-Independent Strategy



Optimal Cost-Independent Strategy

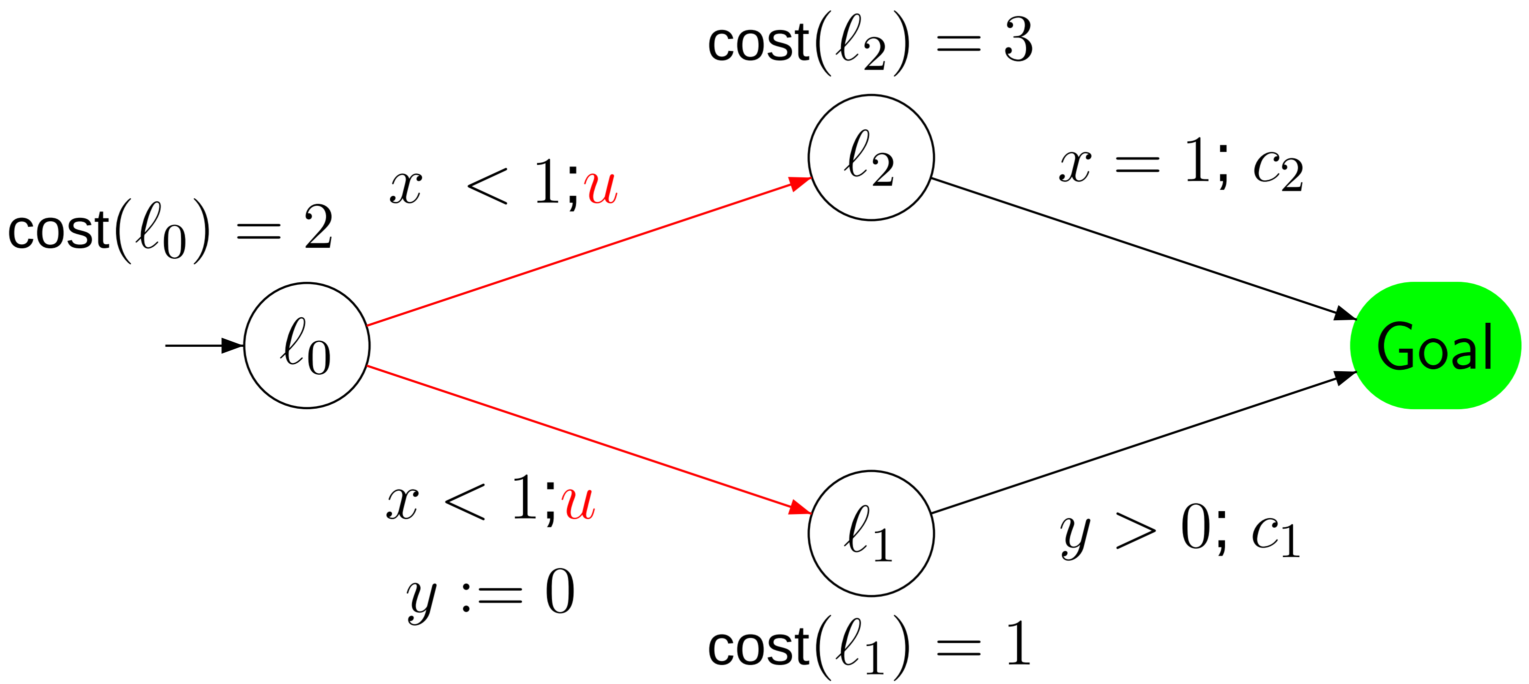
- tagged sets: keep information how to win on W_{i+1}
  - compute $W_{i+1} = \pi(W_i)$ and let $Y = W_{i+1} \setminus W_i$
  - $W_{i+1}^{[c]}$ can reach W_i doing a c
  - $W_{i+1}^{[\lambda]}$ can reach W_i or $\mathsf{cPred}(W_i)$ by time-elapsing
- optimal state-based strategy:
  - on $W_{i+1}^{[c]} \leq W_{i+1}^{[\lambda]}$ do c
  - on $W_{i+1}^{[\lambda]} < W_{i+1}^{[c]}$ do λ

How-To Cost-Independent Strategy



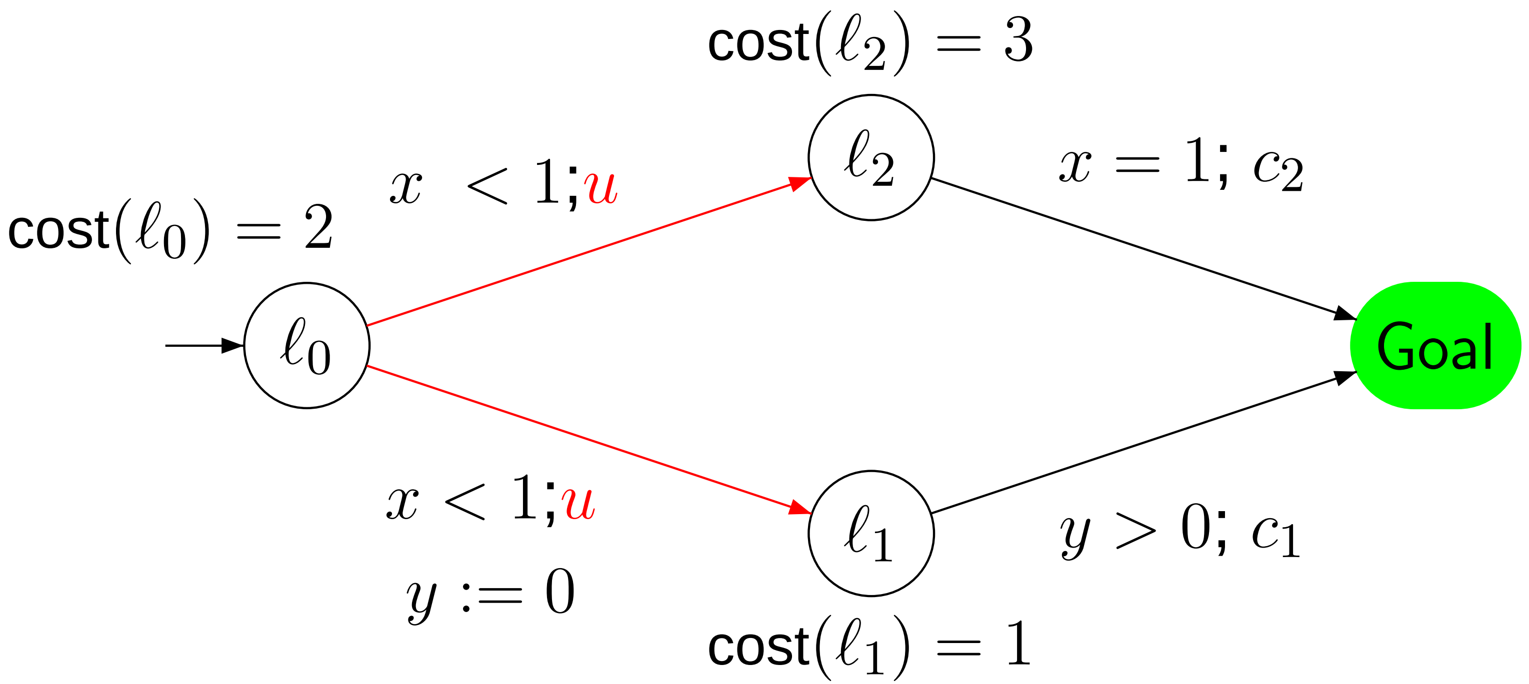
- Optimal cost is 3

How-To Cost-Independent Strategy



- Optimal cost is 3
- Optimal move in $(\ell_1, y > 0) = c_1$, in $(\ell_1, 0) = \lambda$

How-To Cost-Independent Strategy



- Optimal cost is 3
- Optimal move in $(\ell_1, y > 0) = c_1$, in $(\ell_1, 0) = \lambda$
- Optimal strategy: $f^*(\ell_1, 0 < y < \frac{1}{2}) = \lambda$, in $(\ell_1, y \geq \frac{1}{2}) = c_1$ $f^*(\ell_2, x < 1) = \lambda$ and $f^*(\ell_2, x \geq 1) = c_2$