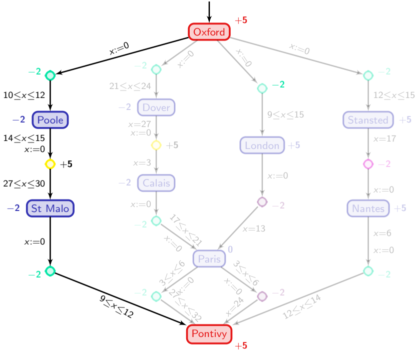
Managing resources in timed systems

Patricia Bouyer-Decitre Uli Fahrenberg
Nicolas Markey Kim G. Larsen Jirí Srba

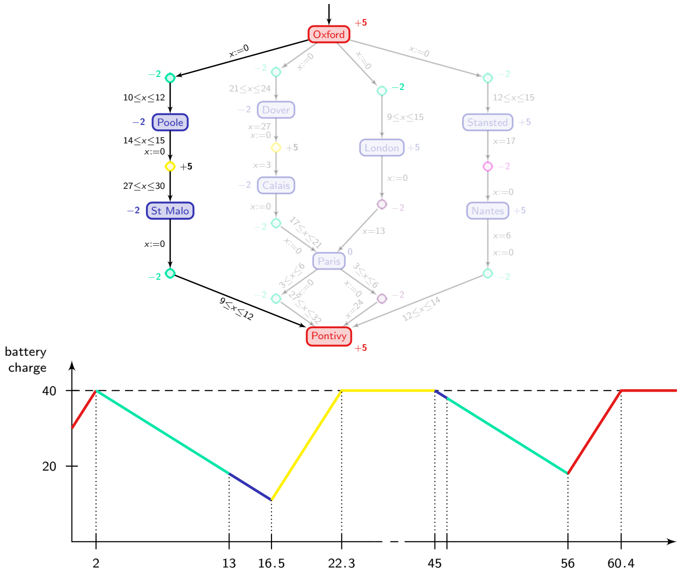
LSV, CNRS & ENS Cachan, France Aalborg University, Denmark

Dagstuhl seminar – January 2010

Can I work with my computer all the way?



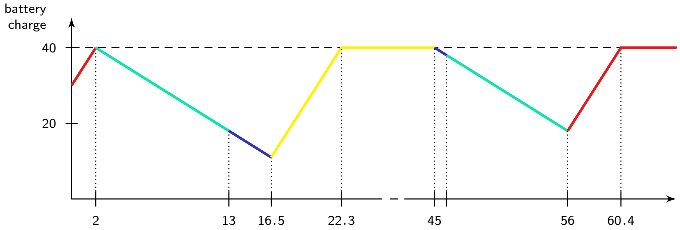
Can I work with my computer all the way?



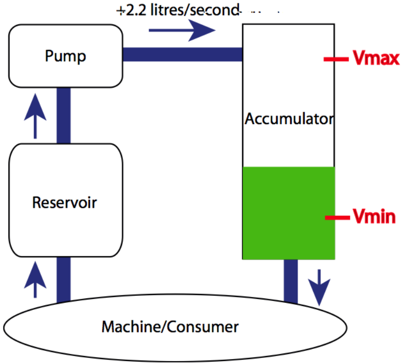
Can I work with my computer all the way?

Energy is not only consumed, but can be regained.

$\leadsto$ the aim is to continuously satisfy some energy constraints.



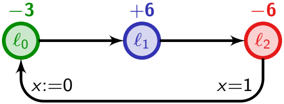
An oil pump control system



[CJL+09] Cassez, Jessen, Larsen, Raskin, Reynier. Automatic synthesis of robust and optimal controllers - An industrial case study *(HSCC'09)*.

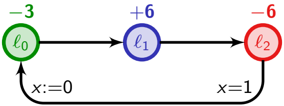
An example of resource management

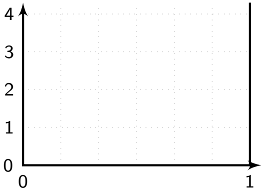
Globally $(x \leq 1)$



An example of resource management

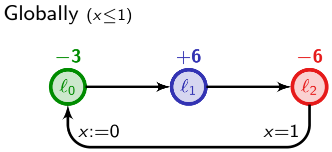
Globally $(x \leq 1)$

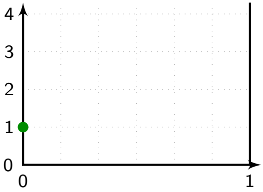




- Lower-bound problem: can we stay above 0?

An example of resource management

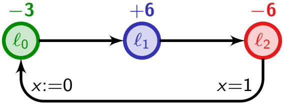


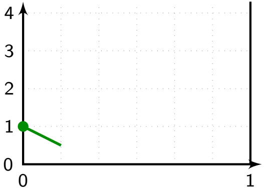


- Lower-bound problem: can we stay above 0?

An example of resource management

Globally $(x \leq 1)$

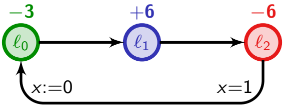


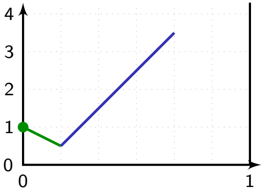


- Lower-bound problem: can we stay above 0?

An example of resource management

Globally $(x \leq 1)$

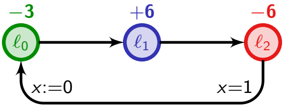


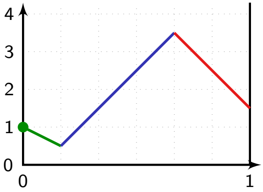


- Lower-bound problem: can we stay above 0?

An example of resource management

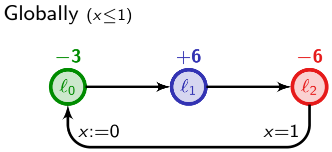
Globally $(x \leq 1)$

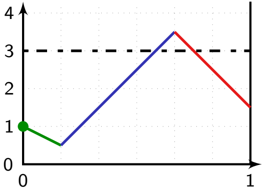




- Lower-bound problem: can we stay above 0?

An example of resource management

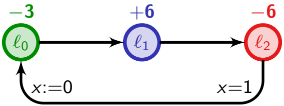


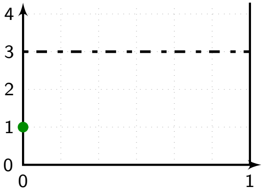


- Lower-bound problem: can we stay above 0?

An example of resource management

Globally $(x \leq 1)$

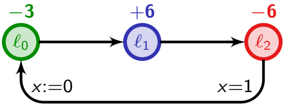


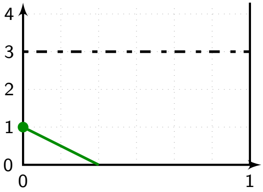


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

Globally $(x \leq 1)$

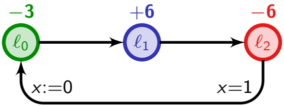


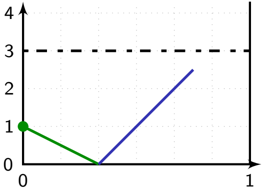


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

Globally $(x \leq 1)$

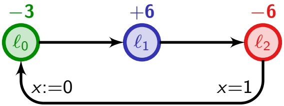


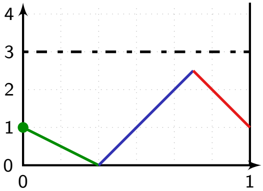


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

Globally $(x \leq 1)$

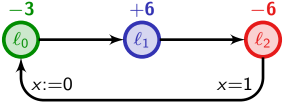


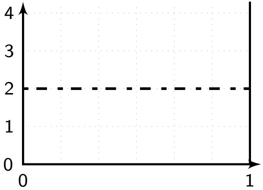


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

Globally $(x \leq 1)$

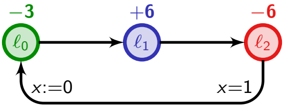


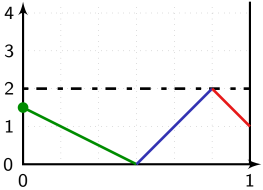


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

Globally $(x \leq 1)$

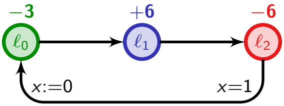


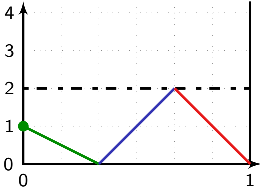


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

Globally $(x \leq 1)$

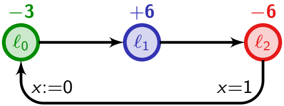


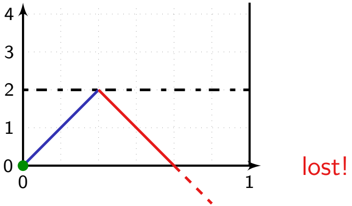


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

Globally $(x \leq 1)$

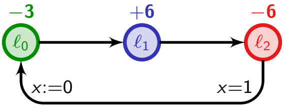


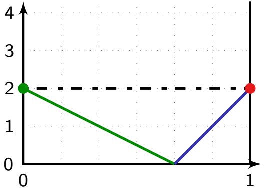


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

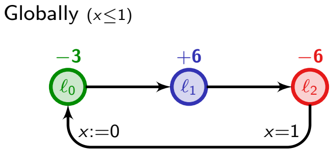
Globally $(x \leq 1)$

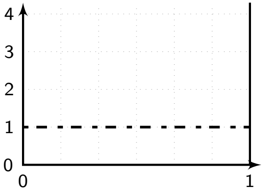




- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

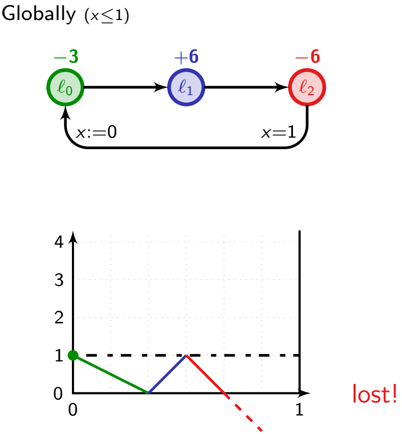
An example of resource management





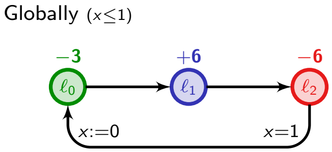
- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

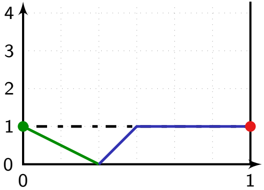
An example of resource management



- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

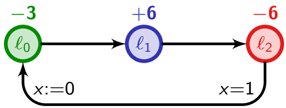


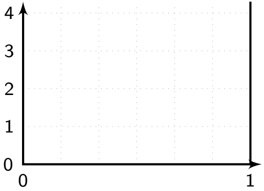


- Lower-bound problem
- Lower-upper-bound problem
- Lower-weak-upper-bound problem: can we "weakly" stay within bounds?

An example of resource management

Globally $(x \leq 1)$





- Lower–bound problem $\rightsquigarrow$ **L**
- Lower-upper-bound problem $\rightsquigarrow$ **L+U**
- Lower-weak-upper-bound problem $\rightsquigarrow$ **L+W**

Only partial results so far

0 clock!	**exist. problem**	**univ. problem**	**games**
L	$\in$ PTIME	$\in$ PTIME	$\in$ UP $\cap$ co-UP PTIME-hard
L+W	$\in$ PTIME	$\in$ PTIME	$\in$ NP $\cap$ co-NP PTIME-hard
L+U	$\in$ PSPACE NP-hard	$\in$ PTIME	EXPTIME-c.

Only partial results so far

1 clock	**exist. problem**	**univ. problem**	**games**
L	$\in$ PTIME	$\in$ PTIME	?
L+W	$\in$ PTIME	$\in$ PTIME	?
L+U	?	?	undecidable

Only partial results so far

n clocks	**exist. problem**	**univ. problem**	**games**
L	?	?	?
L+W	?	?	?
L+U	?	?	undecidable

Back to 0 clock

0 clock!	**exist. problem**	**univ. problem**	**games**
L	$\in$ PTIME	$\in$ PTIME	$\in$ UP $\cap$ co-UP PTIME-hard
L+W	$\in$ PTIME	$\in$ PTIME	$\in$ NP $\cap$ co-NP PTIME-hard
L+U	$\in$ PSPACE NP-hard	$\in$ PTIME	EXPTIME-c.

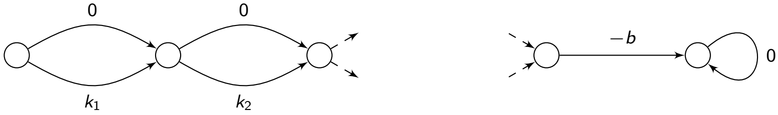
$\leadsto$ PTIME: Bellman-Ford algorithm

Back to 0 clock

0 clock!	exist. problem	univ. problem	games
L	$\in$ PTIME	$\in$ PTIME	$\in$ UP $\cap$ co-UP PTIME-hard
L+W	$\in$ PTIME	$\in$ PTIME	$\in$ NP $\cap$ co-NP PTIME-hard
L+U	$\in$ PSPACE NP-hard	$\in$ PTIME	EXPTIME-c.

$\rightsquigarrow$ PSPACE: guess an infinite path in the graph augmented with the energy level

$\rightsquigarrow$ NP-hardness: encode SUBSET-SUM:



Back to 0 clock

0 clock!	**exist. problem**	**univ. problem**	**games**
L	$\in$ PTIME	$\in$ PTIME	$\in$ UP $\cap$ co-UP PTIME-hard
L+W	$\in$ PTIME	$\in$ PTIME	$\in$ NP $\cap$ co-NP PTIME-hard
L+U	$\in$ PSPACE NP-hard	$\in$ PTIME	EXPTIME-c.

$\rightsquigarrow$ EXPTIME: play the game in the graph augmented with the energy level

$\rightsquigarrow$ EXPTIME-hardness: encode COUNTDOWN-GAME [JLS07]

[JLS07] Jurdziński, Laroussinie, Sproston. Model checking probabilistic timed automata with one or two clocks *(TACAS'07)*.

0 clock: Relation with mean-payoff games

Definition

Mean-payoff games: in a weighted game graph, does there exists a strategy s.t. the mean-cost of any play is nonnegative?

0 clock: Relation with mean-payoff games

Definition

Mean-payoff games: in a weighted game graph, does there exists a strategy s.t. the mean-cost of any play is nonnegative?

Lemma

L-games and **L**$+$**W**-games are determined, and memoryless strategies are sufficient to win.

0 clock: Relation with mean-payoff games

Definition

Mean-payoff games: in a weighted game graph, does there exists a strategy s.t. the mean-cost of any play is nonnegative?

Lemma

L-games and **L**+**W**-games are determined, and memoryless strategies are sufficient to win.

- from mean-payoff games to **L**-games or **L**+**W**-games: play in the same game graph G with initial credit $-M \geq 0$ (where M is the sum of negative costs in G).

0 clock: Relation with mean-payoff games

Definition

Mean-payoff games: in a weighted game graph, does there exists a strategy s.t. the mean-cost of any play is nonnegative?

Lemma

L-games and **L**+**W**-games are determined, and memoryless strategies are sufficient to win.

- from mean-payoff games to **L**-games or **L**+**W**-games: play in the same game graph G with initial credit $-M \geq 0$ (where M is the sum of negative costs in G).

- from **L**-games to mean-payoff games: transform the game as follows:

0 clock: Relation with mean-payoff games

Definition

Mean-payoff games: in a weighted game graph, does there exists a strategy s.t. the mean-cost of any play is nonnegative?

Lemma

L-games and **L**+**W**-games are determined, and memoryless strategies are sufficient to win.

Corollary

Mean-payoff games (and hence parity games) and **L**-games have the same complexity (log-space reducibility).

$\leadsto$ a way to improve complexity of mean-payoff games [DGR09]

[DGR09] Doyen, Gentilini, Raskin. Faster Pseudo-Polynomial Algorithms for Mean-Payoff Games.

What about 1 clock?

1 clock	**exist. problem**	**univ. problem**	**games**
L	$\in$ PTIME	$\in$ PTIME	?
L+W	$\in$ PTIME	$\in$ PTIME	?
L+U	?	?	undecidable

1 clock: **L**- and **L+W**-cases

Idea: delay in the most profitable location

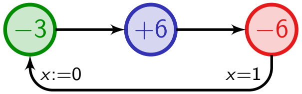
$\rightsquigarrow$ the corner-point abstraction

1 clock: **L**- and **L+W**-cases

Idea: delay in the most profitable location

$\rightsquigarrow$ the corner-point abstraction

Example

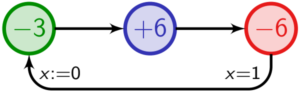


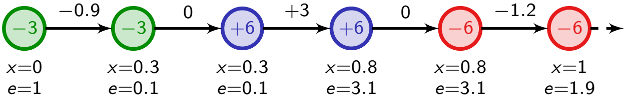
1 clock: **L**- and **L+W**-cases

Idea: delay in the most profitable location

$\rightsquigarrow$ the corner-point abstraction

Example



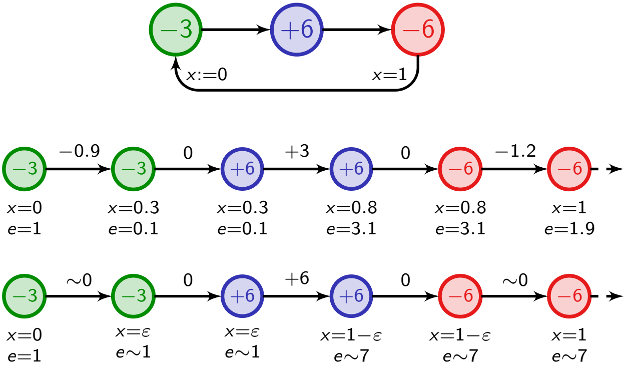


1 clock: **L**- and **L+W**-cases

Idea: delay in the most profitable location

$\rightsquigarrow$ the corner-point abstraction

Example

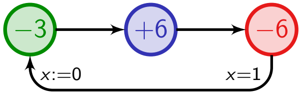


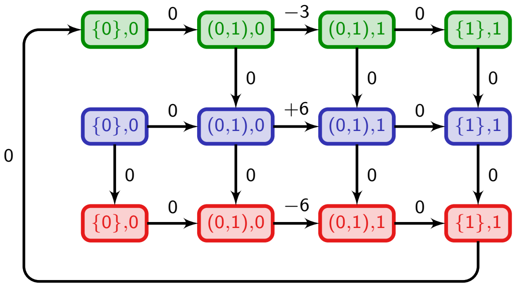
1 clock: **L**- and **L+W**-cases

Idea: delay in the most profitable location

$\rightsquigarrow$ the corner-point abstraction

Example





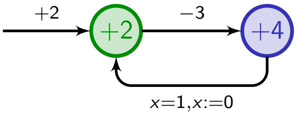
1 clock: **L**- and **L+W**-cases

Idea: delay in the most profitable location

$\rightsquigarrow$ the corner-point abstraction

Remark

The corner-point abstraction is not correct with discrete costs.



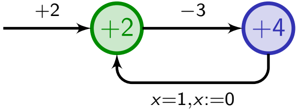
1 clock: **L**- and **L+W**-cases

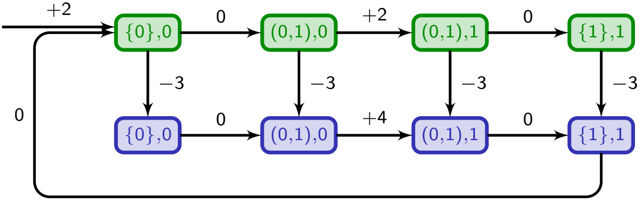
Idea: delay in the most profitable location

$\rightsquigarrow$ the corner-point abstraction

Remark

The corner-point abstraction is not correct with discrete costs.





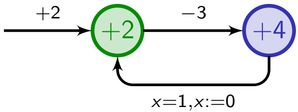
1 clock: **L**- and **L+W**-cases

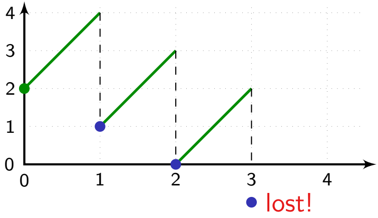
Idea: delay in the most profitable location

$\rightsquigarrow$ the corner-point abstraction

Remark

The corner-point abstraction is not correct with discrete costs.





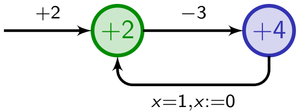
1 clock: **L**- and **L+W**-cases

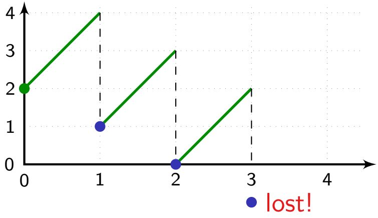
Idea: delay in the most profitable location

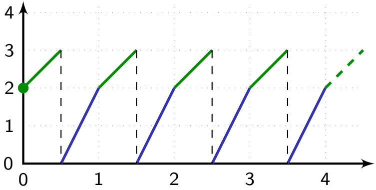
$\rightsquigarrow$ the corner-point abstraction

Remark

The corner-point abstraction is not correct with discrete costs.







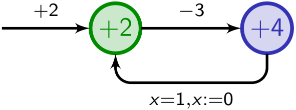
1 clock: **L**- and **L+W**-cases

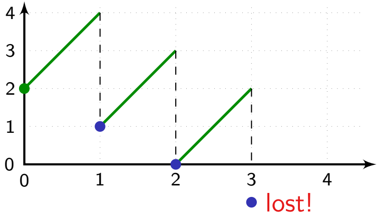
Idea: delay in the most profitable location

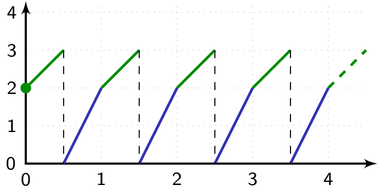
$\rightsquigarrow$ the corner-point abstraction

Remark

The corner-point abstraction is not correct with discrete costs.





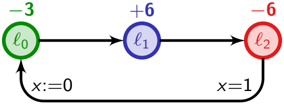


$\rightsquigarrow$ next talk by Nicolas Markey

1 clock: **L+U**-problem and fixpoint computation

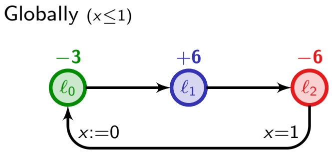
The backward fixpoint computation, which is correct in the limit, does not terminate in general.

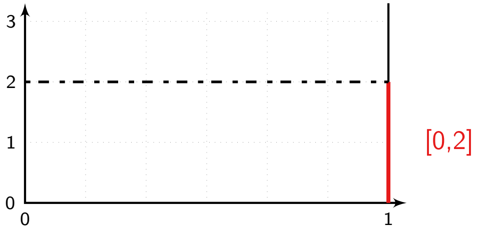
Globally $(x \leq 1)$



1 clock: **L+U**-problem and fixpoint computation

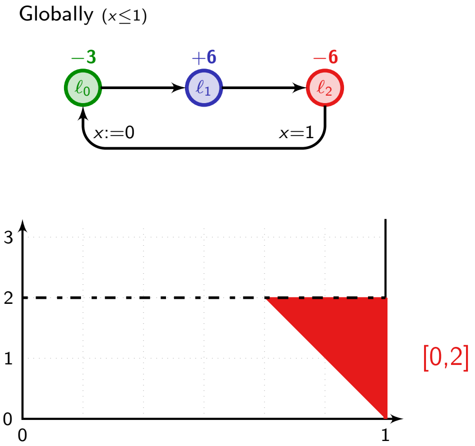
The backward fixpoint computation, which is correct in the limit, does not terminate in general.





1 clock: **L+U**-problem and fixpoint computation

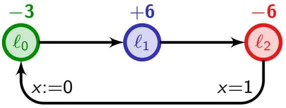
The backward fixpoint computation, which is correct in the limit, does not terminate in general.

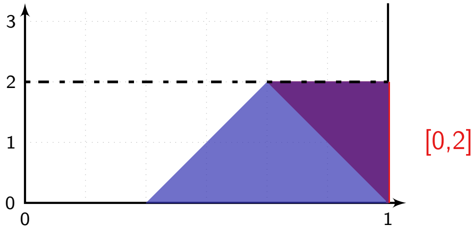


1 clock: **L+U**-problem and fixpoint computation

The backward fixpoint computation, which is correct in the limit, does not terminate in general.

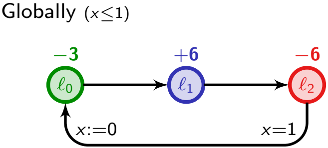
Globally $(x \leq 1)$

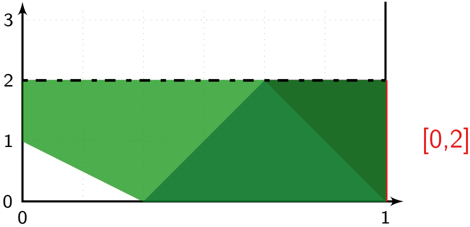




1 clock: **L+U**-problem and fixpoint computation

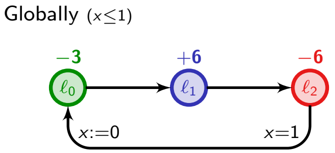
The backward fixpoint computation, which is correct in the limit, does not terminate in general.

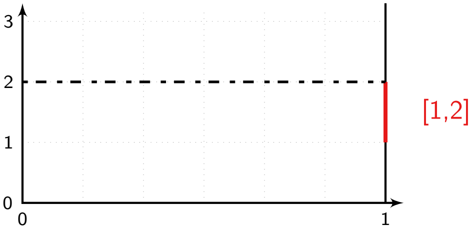




1 clock: **L+U**-problem and fixpoint computation

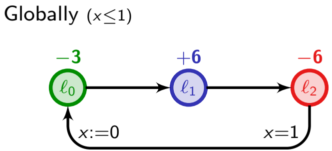
The backward fixpoint computation, which is correct in the limit, does not terminate in general.

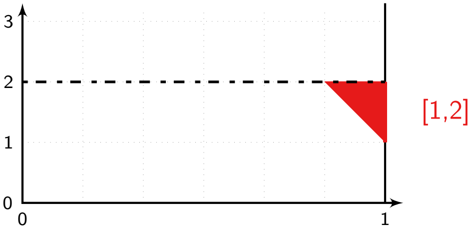




1 clock: **L+U**-problem and fixpoint computation

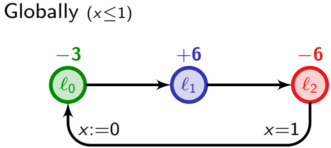
The backward fixpoint computation, which is correct in the limit, does not terminate in general.

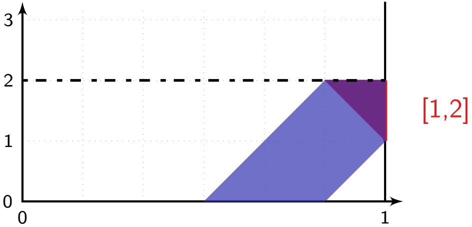




1 clock: **L+U**-problem and fixpoint computation

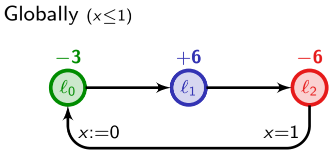
The backward fixpoint computation, which is correct in the limit, does not terminate in general.

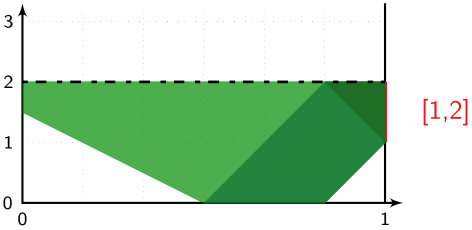




1 clock: **L+U**-problem and fixpoint computation

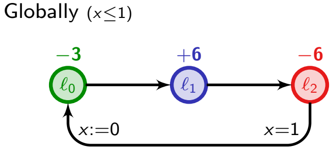
The backward fixpoint computation, which is correct in the limit, does not terminate in general.

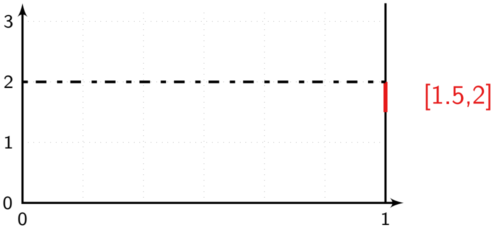




1 clock: **L+U**-problem and fixpoint computation

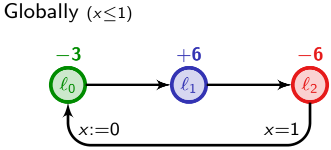
The backward fixpoint computation, which is correct in the limit, does not terminate in general.

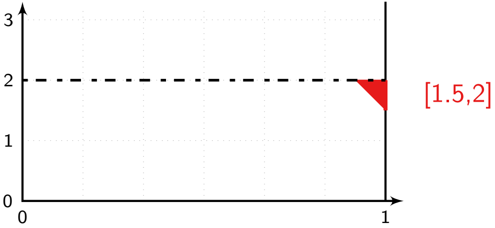




1 clock: **L+U**-problem and fixpoint computation

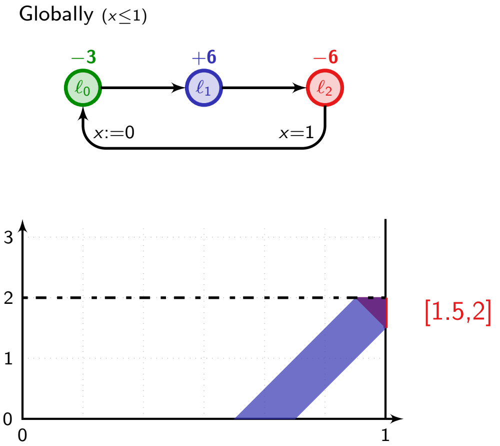
The backward fixpoint computation, which is correct in the limit, does not terminate in general.





1 clock: **L+U**-problem and fixpoint computation

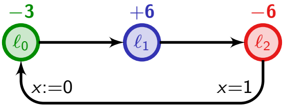
The backward fixpoint computation, which is correct in the limit, does not terminate in general.

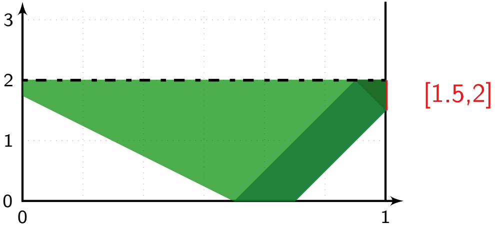


1 clock: **L+U**-problem and fixpoint computation

The backward fixpoint computation, which is correct in the limit, does not terminate in general.

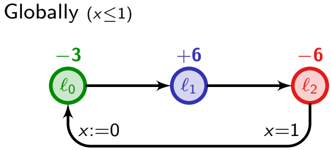
Globally $(x \leq 1)$

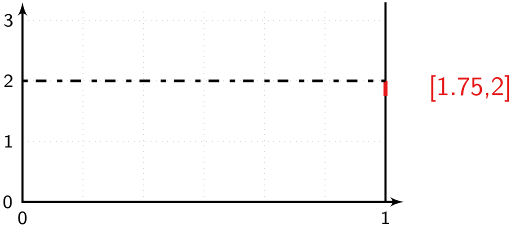




1 clock: **L+U**-problem and fixpoint computation

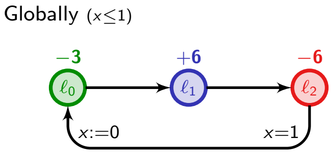
The backward fixpoint computation, which is correct in the limit, does not terminate in general.

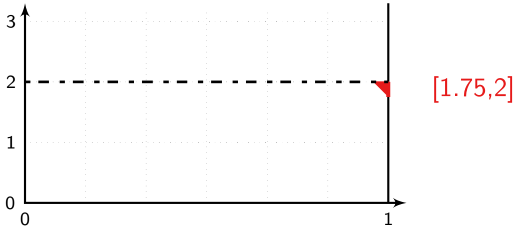




1 clock: **L+U**-problem and fixpoint computation

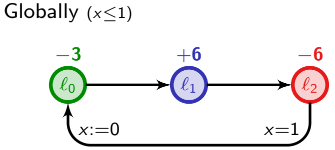
The backward fixpoint computation, which is correct in the limit, does not terminate in general.

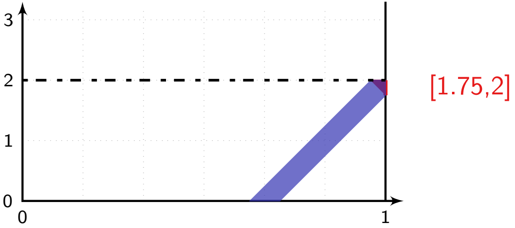




1 clock: **L+U**-problem and fixpoint computation

The backward fixpoint computation, which is correct in the limit, does not terminate in general.

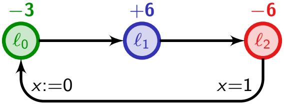


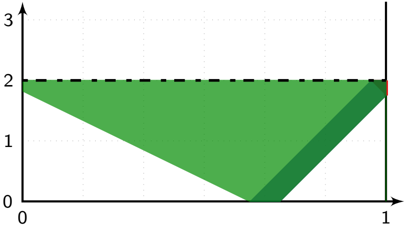


1 clock: **L+U**-problem and fixpoint computation

The backward fixpoint computation, which is correct in the limit, does not terminate in general.

Globally $(x \leq 1)$



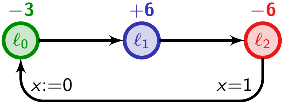


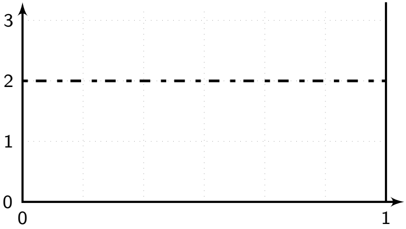
[1.75,2]

1 clock: **L+U**-problem and fixpoint computation

The backward fixpoint computation, which is correct in the limit, does not terminate in general.

Globally $(x \leq 1)$

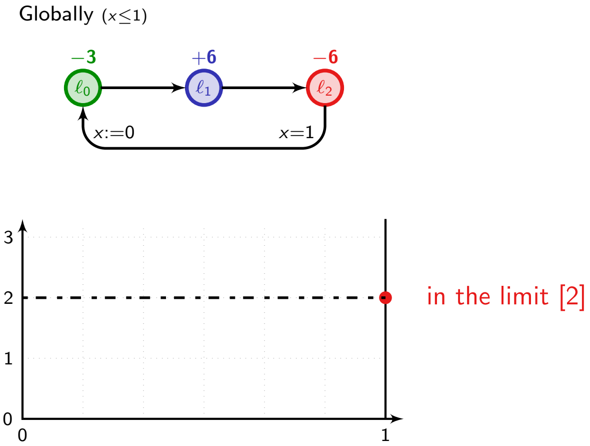




etc...

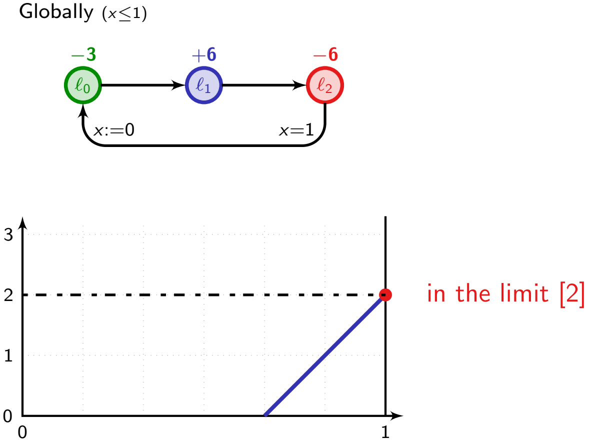
1 clock: **L+U**-problem and fixpoint computation

The backward fixpoint computation, which is correct in the limit, does not terminate in general.



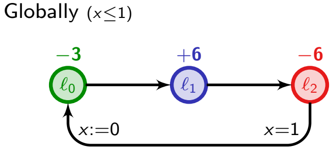
1 clock: **L+U**-problem and fixpoint computation

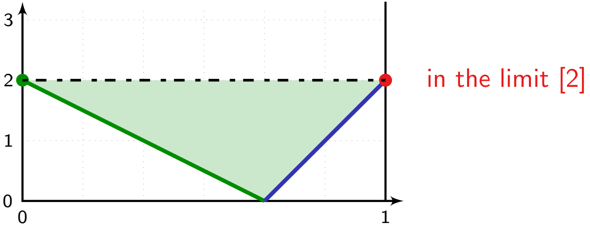
The backward fixpoint computation, which is correct in the limit, does not terminate in general.



1 clock: **L+U**-problem and fixpoint computation

The backward fixpoint computation, which is correct in the limit, does not terminate in general.





1 clock: **L+U**-games

Theorem

The single-clock **L**+**U**-games are undecidable.

1 clock: **L+U**-games

Theorem

The single-clock **L**+**U**-games are undecidable.

We encode the behaviour of a two-counter machine:

- each instruction is encoded as a module;
- the values c_1 and c_2 of the counters are encoded by the energy level

$$e = 5 - \frac{1}{2^{c_1} \cdot 3^{c_2}}$$

when entering the corresponding module.

1 clock: **L+U**-games

Theorem

The single-clock **L**+**U**-games are undecidable.

We encode the behaviour of a two-counter machine:

- each instruction is encoded as a module;
- the values c_1 and c_2 of the counters are encoded by the energy level

$$e = 5 - \frac{1}{2^{c_1} \cdot 3^{c_2}}$$

when entering the corresponding module.

There is an infinite execution in the two-counter machine iff there is a strategy in the single-clock timed game under which the energy level remains between 0 and 5.

1 clock: **L+U**-games

Theorem

The single-clock **L**+**U**-games are undecidable.

We encode the behaviour of a two-counter machine:

- each instruction is encoded as a module;
- the values c_1 and c_2 of the counters are encoded by the energy level

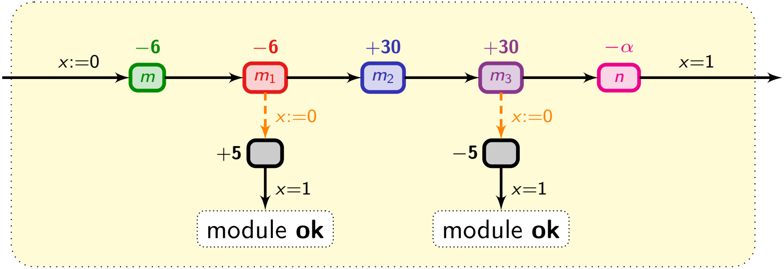
$$e = 5 - \frac{1}{2^{c_1} \cdot 3^{c_2}}$$

  when entering the corresponding module.

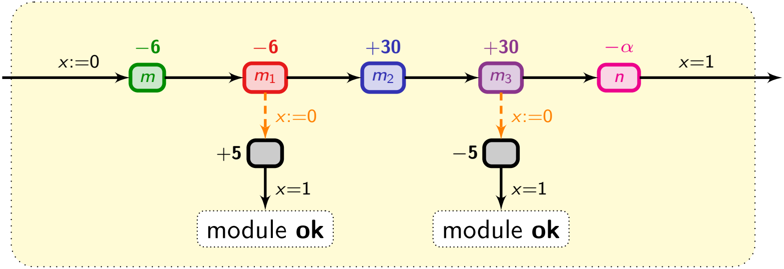
There is an infinite execution in the two-counter machine iff there is a strategy in the single-clock timed game under which the energy level remains between 0 and 5.

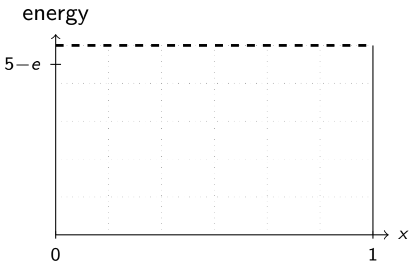
$\rightsquigarrow$ We present a generic construction for incrementing/decrementing the counters.

Generic module for incrementing/decrementing

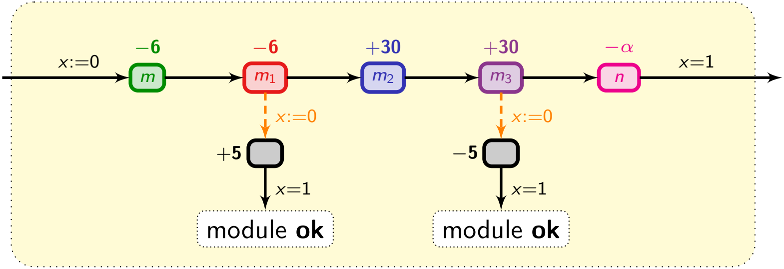


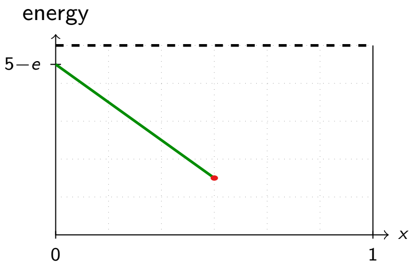
Generic module for incrementing/decrementing



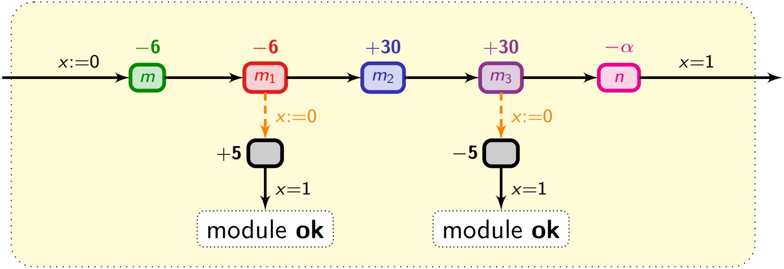


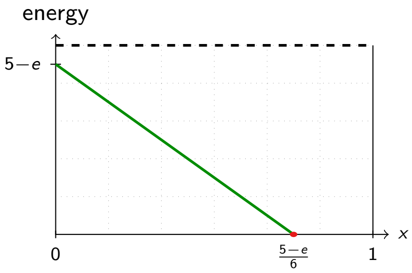
Generic module for incrementing/decrementing



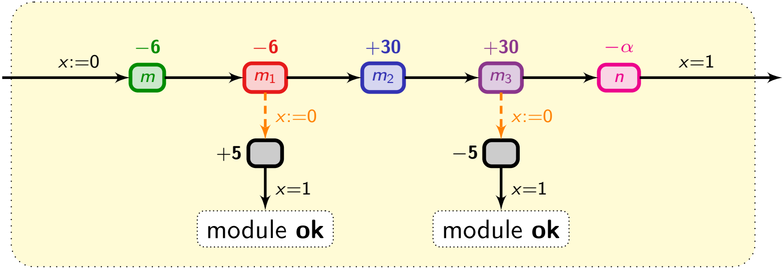


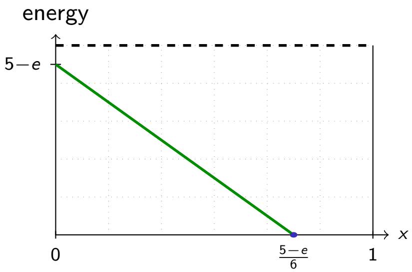
Generic module for incrementing/decrementing



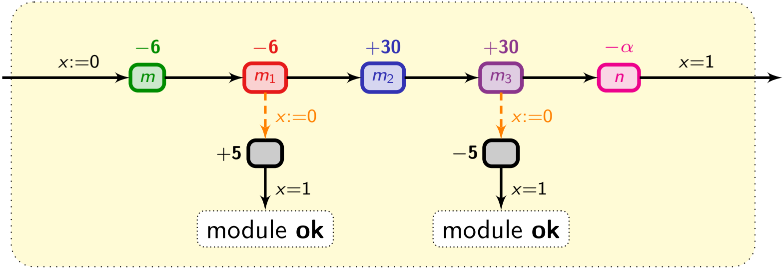


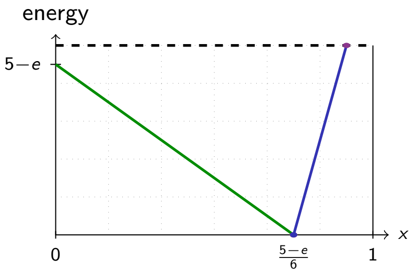
Generic module for incrementing/decrementing



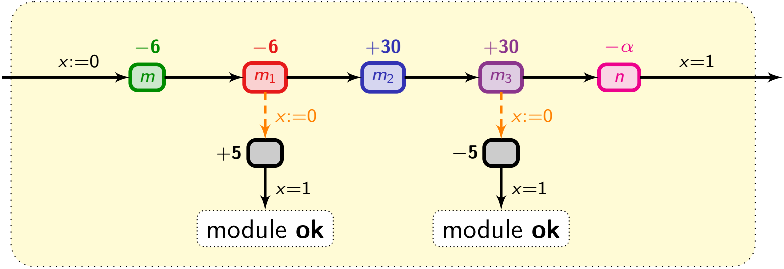


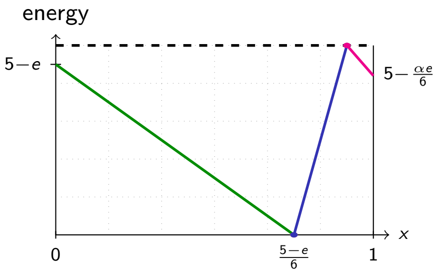
Generic module for incrementing/decrementing



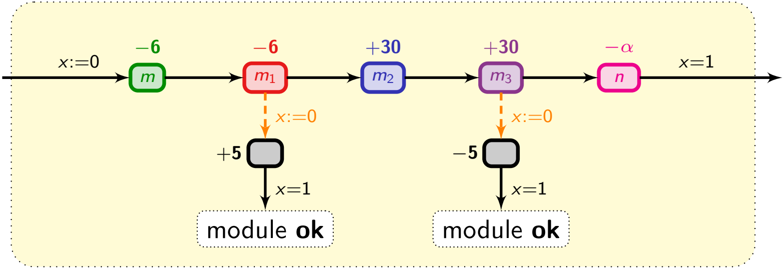


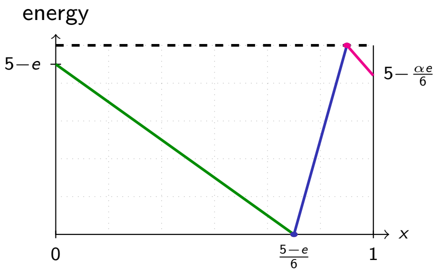
Generic module for incrementing/decrementing





Generic module for incrementing/decrementing





- α=3: increment c_1
- α=2: increment c_2
- α=12: decrement c_1
- α=18: decrement c_2

Conclusion & ongoing/future work

- Extension of weighted/priced (timed) automata with negative costs:
  - three natural problems related to the management of resources;
  - reasonable complexity in the untimed case;
  - undecidable for 1-clock games.

Conclusion & ongoing/future work

- Extension of weighted/priced (timed) automata with negative costs:
  - three natural problems related to the management of resources;
  - reasonable complexity in the untimed case;
  - undecidable for 1-clock games.

- Many open problems:
  - in the untimed case:
    - existential problem with interval constraints: the problem is related to reachability in 2-clock timed automata;
    - games with lower-bound constraint: the problem is equivalent to the mean-payoff game problem.
  - in the timed case:
    - many open questions for the 1-clock case, and no results for the general case.

Conclusion & ongoing/future work

- Extension of weighted/priced (timed) automata with negative costs:
  - three natural problems related to the management of resources;
  - reasonable complexity in the untimed case;
  - undecidable for 1-clock games.

- Many open problems:
  - in the untimed case:
    - existential problem with interval constraints: the problem is related to reachability in 2-clock timed automata;
    - games with lower-bound constraint: the problem is equivalent to the mean-payoff game problem.
  - in the timed case:
    - many open questions for the 1-clock case, and no results for the general case.

- Other cost functions? $\leadsto$ see next talk