Timed Games with Window Parity Objectives

James C. A. Main [1]   Mickael Randour [1]   Jeremy Sproston [2]

[1]UMONS – Université de Mons and F.R.S.-FNRS, Belgium

[2]Università degli Studi di Torino, Italy

Journées du GT Vérif – November 18, 2021

Introduction

Window objectives have been studied in discrete-time settings:

- in turn-based games with mean-payoff and total-payoff objectives [CDRR15];
- in turn-based games with parity objectives [BHR16];
- in Markov decision processes for parity and mean-payoff objectives [BDOR20].

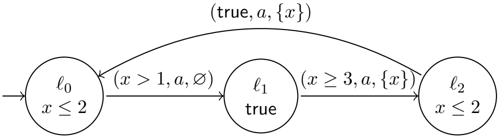
We have extended window parity objectives to a continuous-time setting, for timed automata and timed games $\rightsquigarrow$ time is not measured as steps !

Intuition of window parity objectives

A window parity objective for a fixed bound λ requires that, in all configurations occurring in a play, we see a good window for the parity objective, i.e., a time frame of size less than λ such that the smallest priority in this time frame is even.

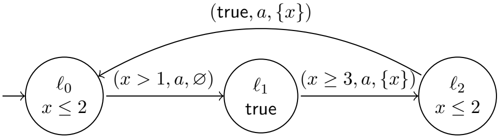
Timed automata

- Timed automata [AD94] are used to model real-time systems.
- The elapse of time is measured by a finite number of clock variables, or clocks, that progress at the same rate.
- Clock constraints are conjunctions of conditions of the form $x \leq c$, $x < c$, $x \geq c$ and $x > c$ where x is a clock and c a natural number.



Timed automata

- Timed automata [AD94] are used to model real-time systems.
- The elapse of time is measured by a finite number of clock variables, or clocks, that progress at the same rate.
- Clock constraints are conjunctions of conditions of the form $x \leq c$, $x < c$, $x \geq c$ and $x > c$ where x is a clock and c a natural number.



- Timed automata consist of:
  - a finite set of locations constrained by invariants with a distinguished initial location ℓ_{init} and
  - a finite set of edges labeled by guards, actions and clocks to reset.

Timed automata

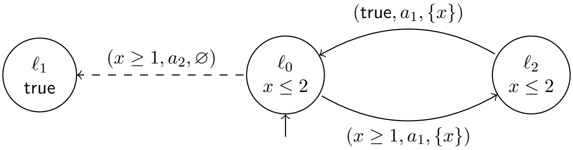
Semantics

A timed automaton gives rise to an uncountable transition system.

- States of this transition system are pairs of locations and clock valuations (i.e., mappings $C \to \mathbb{R}^{\geq 0}$). The initial state is $(\ell_{\text{init}}, \mathbf{0}^C)$.
- Moves are pairs (d, a) where $d \in \mathbb{R}^{\geq 0}$ is a delay and a is an action of the timed automaton or a special standby action $\bot$.
- Transitions are constrained by the invariants and guards of the timed automaton.
  - One can wait in a location of a timed automaton as long as its invariant is satisfied.
  - One can traverse an edge of a timed automaton after a delay d if the invariant of the current location and the guard of the edge are satisfied after d time units, and the invariant of the target location is satisfied after resetting the clocks specified on the edge.

Timed games

- We consider two-player games played on timed automata.
- A timed game is given by a timed automaton and a partition of the actions of the timed automaton in $\mathcal{P}_1$ actions and $\mathcal{P}_2$ actions.
- These games are concurrent: at each round, both players present a move and the play proceeds following a fastest move – a transition is chosen non-deterministically if both players present moves with the same delay.



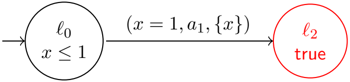
- Example 1: $(\ell_0, 0)\,((1, a_1), (1, a_2))\,(\ell_1, 1)$
- Example 2: $(\ell_0, 0)\,((1, a_1), (1, a_2))\,(\ell_2, 0)$

Timed games

- Plays are non-terminating: a play is an infinite sequence of alternating states of the transition system underlying the timed automaton and pairs consisting of $\mathcal{P}_1$ and $\mathcal{P}_2$ moves.
- A strategy for $\mathcal{P}_i$ is a function mapping histories (i.e., finite prefixes of plays) to moves of $\mathcal{P}_i$.
- An objective is a set of plays that represents the specification to be enforced.

The passage of time in timed games

- It is possible to have a play in which a finite amount of time passes.
  - Example: $(\ell_0, 0)((\frac{1}{2}, \bot), (\frac{1}{2}, \bot))(\ell_0, \frac{1}{2})((\frac{1}{4}, \bot), (\frac{1}{4}, \bot))(\ell_0, \frac{3}{4}) \dots$

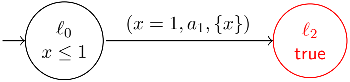


- Plays in which the sum of delays converges are called time-convergent.
- Otherwise, a play is referred to as time-divergent.

Time-convergent plays are not physically meaningful...

The passage of time in timed games

- It is possible to have a play in which a finite amount of time passes.
  - Example: $(\ell_0, 0)((\frac{1}{2}, \bot), (\frac{1}{2}, \bot))(\ell_0, \frac{1}{2})((\frac{1}{4}, \bot), (\frac{1}{4}, \bot))(\ell_0, \frac{3}{4}) \ldots$



- Plays in which the sum of delays converges are called time-convergent.
- Otherwise, a play is referred to as time-divergent.

Time-convergent plays are not physically meaningful...

$\rightsquigarrow$ we do not want $\mathcal{P}_1$ to enforce his objective by making time converge !

Winning in timed games

- Due to the phenomenon of time-convergence, we distinguish objectives and winning conditions, following [dAFH+03].
- Given an objective, we say a play belongs to its associated winning condition if one of the two following conditions is fulfilled:
  - the play is time-divergent and satisfies the objective;
  - the play is time-convergent and from some point on, transitions in the play cannot be achieved by $\mathcal{P}_1$'s moves.
- We say a strategy is winning from some initial state if all plays starting in this state consistent with the strategy satisfy the winning condition.

Realizability problem

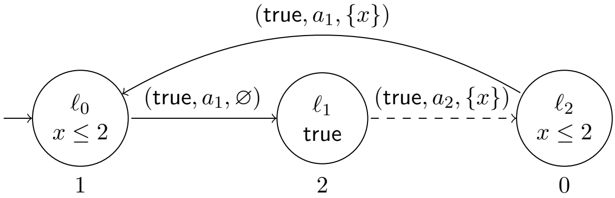
Given an objective, check whether $\mathcal{P}_1$ has a winning strategy from the initial state.

Objectives of interest

- Safety objective: for a set of locations F, the safety objective over F, denoted by $\mathsf{Safe}(F)$, consists of sequences of states along which no location in F ever appears.
- Co-Büchi objective: for a set of locations F, the co-Büchi objective over F, denoted by $\mathsf{coBüchi}(F)$, consists of sequences of states along which no location in F appears infinitely often.
- Parity objective: given a priority function p mapping a non-negative integer to locations, consists of sequences of states along which the smallest priority appearing infinitely often is even.

A motivation for windows

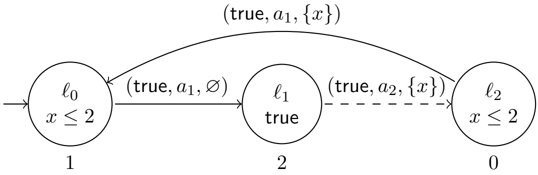
- For the classical parity objective, there are no timing constraints between odd priorities and smaller even priorities.



- Through the window mechanism, one can specify such timing constraints.

Good windows

- The window objectives are based on the notion of good windows.
- Fix a bound λ on the length of windows. A good window for the parity objective is a window in which:
  - strictly less than λ time units elapse and
  - the smallest priority appearing in the window is even.



Examples for $\lambda = 2$:

- $\left[(\ell_0, 0)((1, a_1), (2, \bot))(\ell_1, 1)((5, \bot), (\frac{1}{2}, a_2))(\ell_2, 0)((0, a_1), (1, \bot))\right]^\omega$
  $\rightsquigarrow$ good window at the start
- $\left[(\ell_0, 0)((1, a_1), (2, \bot))(\ell_1, 1)((5, \bot), (\frac{3}{2}, a_2))(\ell_2, 0)((0, a_1), (1, \bot))\right]^\omega$
  $\rightsquigarrow$ bad window at the start

Some convenient notation

- For two moves $m^{(1)} = (d^{(1)}, a^{(1)})$, $m^{(2)} = (d^{(2)}, a^{(2)})$, write

$$\mathsf{delay}(m^{(1)}, m^{(2)}) = \min\{d^{(1)}, d^{(2)}\}.$$

- Let $\pi = (\ell_0, v_0)(m_0^{(1)}, m_0^{(2)})(\ell_1, v_1)\dots$ be a play. Set, for any $n \in \mathbb{N}$ and $d \leq \mathsf{delay}(m_n^{(1)}, m_n^{(2)})$, $\pi_{n\to}^{+d}$ to be the play

$$(\ell_n, v_n{+}d)(m_n^{(1)}{-}d, m_n^{(2)}{-}d)(\ell_{n+1}, v_{n+1})(m_{n+1}^{(1)}, m_{n+1}^{(2)})(\ell_{n+2}, v_{n+2})\dots$$

Good windows

Let $\pi = (\ell_0, v_0)(m_0^{(1)}, m_0^{(2)})(\ell_1, v_1)\dots$ be a play.

- Timed good window parity objective: the window at the start of the play is good. Formally, we define $\pi \in \mathsf{TGW}(\lambda)$ if and only if

$$\exists\, n,\ \left(\min_{0\leq k\leq n} p(\ell_k)\right) \bmod 2 = 0 \wedge \sum_{k=0}^{n-1} \mathsf{delay}(m_k^{(1)}, m_k^{(2)}) < \lambda.$$

- We say that the window opened at some step j closes at step n if n satisfies

$$(\min_{j\leq k\leq n} p(\ell_k)) \bmod 2 = 0 \wedge \forall\, j \leq n' < n,\ (\min_{j\leq k\leq n'} p(\ell_k)) \bmod 2 = 1.$$

- If a window does not close in strictly less than λ time units, we say that this window is bad.

Timed window parity objectives

Let $\pi = (\ell_0, v_0)(m_0^{(1)}, m_0^{(2)})(\ell_1, v_1)\dots$ be a play.

- Direct timed window parity objective: there is a good window at all steps. We say that $\pi \in \mathsf{DTW}(\lambda)$ if and only if

$$\forall\, n,\ \forall\, d \in [0, \mathsf{delay}(m_n^{(1)}, m_n^{(2)})],\ \pi_{n\to}^{+d} \in \mathsf{TGW}(\lambda).$$

Timed window parity objectives

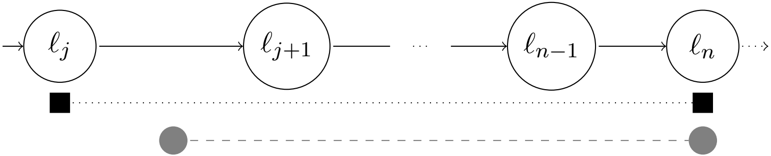
Let $\pi = (\ell_0, v_0)(m_0^{(1)}, m_0^{(2)})(\ell_1, v_1)\ldots$ be a play.

- Direct timed window parity objective: there is a good window at all steps. We say that $\pi \in \mathsf{DTW}(\lambda)$ if and only if

$$\forall\, n,\ \forall\, d \in [0, \mathsf{delay}(m_n^{(1)}, m_n^{(2)})],\ \pi_{n\rightarrow}^{+d} \in \mathsf{TGW}(\lambda).$$

  The above is equivalent to the simpler statement:

$$\forall\, n,\ \pi_{n\rightarrow}^{+0} \in \mathsf{TGW}(\lambda).$$



Timed window parity objectives

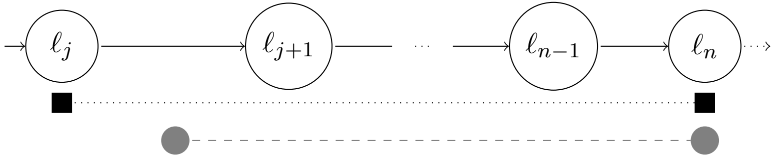
Let $\pi = (\ell_0, v_0)(m_0^{(1)}, m_0^{(2)})(\ell_1, v_1)\dots$ be a play.

- Direct timed window parity objective: there is a good window at all steps. We say that $\pi \in \mathsf{DTW}(\lambda)$ if and only if

  $$\forall\, n,\ \forall\, d \in [0, \mathsf{delay}(m_n^{(1)}, m_n^{(2)})],\ \pi_{n\rightarrow}^{+d} \in \mathsf{TGW}(\lambda).$$

  The above is equivalent to the simpler statement:

  $$\forall\, n,\ \pi_{n\rightarrow}^{+0} \in \mathsf{TGW}(\lambda).$$



- Timed window parity objective: the direct window parity objective holds from some point on; $\pi \in \mathsf{TW}(\lambda)$ holds if and only if

  $$\exists\, n,\ \pi_{n\rightarrow}^{+0} \in \mathsf{DTW}(\lambda).$$

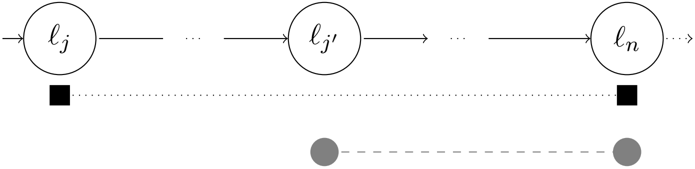
Inductive property of windows

The key to our reduction-based algorithm is the inductive property of windows.

Inductive property of windows

Along all plays of a timed game, for all j, if the window opened at step j closes at step n in strictly less than λ time units, then for all $j \leq j' \leq n$, the window opened at step j' is good.

$$p(\ell_n) = \min_{j \leq k \leq n} p(\ell_k)$$



Inductive property of windows

$\rightsquigarrow$ The inductive property implies that it suffices to keep track of one window at a time.

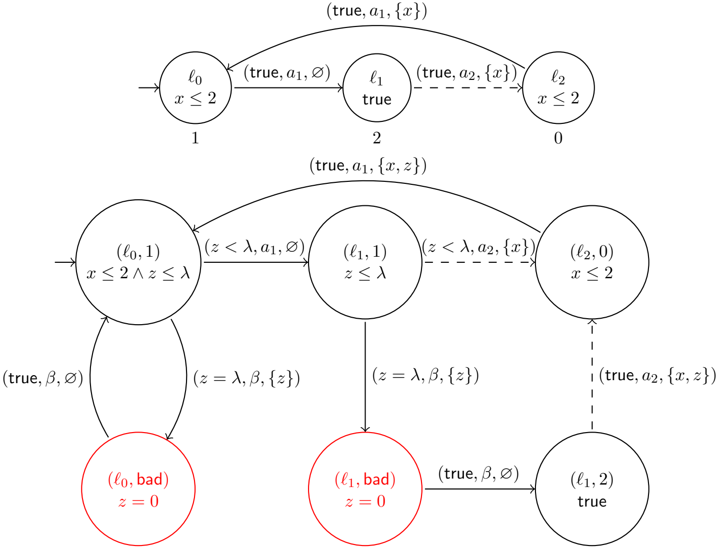
- One can reduce the realizability problem for the direct timed window parity objective to the realizability problem for the safety objective.
- One can reduce the realizability problem for the timed window parity objective to the realizability problem for the co-Büchi objective.

Reduction

- We expand timed automata to include information on the current window use this information to detect bad windows.
- We expand locations to encode the current lowest priority of the window or a special value bad to be avoided.
- We introduce a new clock z to measure how long the current window has been open.
- We change guards and invariants so that bad locations are visited whenever a bad window is witnessed.
- For each player, we add a new action to enter and exit bad locations, written β_1 and β_2.

Key ingredient of the reduction $\rightsquigarrow$ time-divergence

Example of the reduction



Correctness of the reduction

Correctness can be proven using two mappings: an expansion mapping Ex and a projection mapping Pr over histories and plays:

- Ex maps plays of a timed game to plays of its expansion;
- Pr maps plays of an expanded timed game to plays of the original one.

Theorem

For all time-divergent plays π, π satisfies $\mathsf{DTW}(\lambda)$ *(resp.* $\mathsf{TW}(\lambda)$*) if and only if* $\mathsf{Ex}(\pi)$ *satisfies* $\mathsf{Safe}(\mathsf{Bad})$ *(resp.* $\mathsf{coBüchi}(\mathsf{Bad})$*).*

Theorem

For all time-divergent initial plays π of an expanded timed game, π satisfies $\mathsf{Safe}(\mathsf{Bad})$ *(resp.* $\mathsf{coBüchi}(\mathsf{Bad})$*) if and only if* $\mathsf{Pr}(\pi)$ *satisfies* $\mathsf{DTW}(\lambda)$ *(resp.* $\mathsf{TW}(\lambda)$*).*

Correctness of the reduction

The mappings Ex and Pr can be used to translate winning strategies between a timed game and its expansion.

- The expansion mapping can be used to translate strategies of an expanded timed game to strategies of the original timed game.

  Roughly: $\bar{\sigma}$ translated to $\bar{\sigma} \circ \mathsf{Ex}$

  To obtain a well-defined strategy, we replace any β_1 by $\bot$.

- The projection mapping can be used to translate strategies of a timed game to strategies of its expansion.

  Roughly: σ translated to $\sigma \circ \mathsf{Pr}$

  To obtain a well-defined strategy, we use (d, β_1) for some d to replace illegal moves and $(0, \beta_1)$ when in a bad location.

Correctness of the reduction

For a timed game $\mathcal{G}$, let $\mathcal{G}(\lambda)$ denote its expansion.

Theorem

Let s_{init} be the initial state of $\mathcal{G}$ and $\bar{s}_{\mathsf{init}}$ be the initial state of $\mathcal{G}(\lambda)$.

- *There is a winning strategy σ for $\mathcal{P}_1$ for the objective $\mathsf{DTW}(\lambda)$ from s_{init} in $\mathcal{G}$ if and only if there is a winning strategy $\bar{\sigma}$ for $\mathcal{P}_1$ for the objective $\mathsf{Safe}(\mathsf{Bad})$ from $\bar{s}_{\mathsf{init}}$ in $\mathcal{G}(\lambda)$.*
- *There is a winning strategy σ for $\mathcal{P}_1$ for the objective $\mathsf{TW}(\lambda)$ from s_{init} in $\mathcal{G}$ if and only if there is a winning strategy $\bar{\sigma}$ for $\mathcal{P}_1$ for the objective $\mathsf{coBüchi}(\mathsf{Bad})$ from $\bar{s}_{\mathsf{init}}$ in $\mathcal{G}(\lambda)$.*

Multi-dimensional objectives and complexity

- The reduction can be adapted for conjunctions of direct timed window parity objectives and conjunctions of timed window parity objectives.
- By the inductive property, we need only keep track of one window per dimension.
- The construction is similar: locations are expanded with vectors of priorities and one new clock per objective is introduced.

Complexity of the reduction-based algorithm

- In the single-dimensional case, we have a polynomial-time reduction to the realizability problem for timed safety or co-Büchi games, which is an EXPTIME-complete problem.
- In the multi-dimensional case, we also have an EXPTIME complexity for our algorithm.

Verification of timed automata

Verification problem for timed automata

Given an objective, check whether all time-divergent paths of the timed automata satisfy the objective.

- We can use the same reduction as for timed games to handle verification of timed automata with window parity objectives.
- The verification problem for the one-dimensional direct/non-direct timed window parity objective can be reduced in polynomial time to the verification problem for the safety/co-Büchi objective.

Complexity overview

One can show that realizability in timed games and verification of timed automata with safety objectives can be respectively reduced to realizability in timed games and verification in timed automata with timed window parity objectives. This yields EXPTIME-hardness in the case of games and PSPACE-hardness in the case of automata.

Complexity summary

	Single dimension	Multiple dimensions
Timed automata	PSPACE-complete	PSPACE-complete
Timed games	EXPTIME-complete	EXPTIME-complete
Games (untimed) [BHR16]	P-complete	EXPTIME-complete

References I

Rajeev Alur and David L. Dill.
A theory of timed automata.
Theor. Comput. Sci., 126(2):183–235, 1994.

Thomas Brihaye, Florent Delgrange, Youssouf Oualhadj, and Mickael Randour.
Life is random, time is not: Markov decision processes with window objectives.
Log. Methods Comput. Sci., 16(4), 2020.

Véronique Bruyère, Quentin Hautem, and Mickael Randour.
Window parity games: an alternative approach toward parity games with time bounds.
In Domenico Cantone and Giorgio Delzanno, editors, Proceedings of the Seventh International Symposium on Games, Automata, Logics

References II

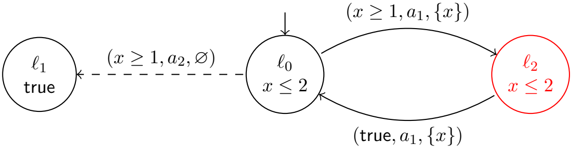
and Formal Verification, GandALF 2016, Catania, Italy, 14-16 September 2016, volume 226 of EPTCS, pages 135–148, 2016.

Krishnendu Chatterjee, Laurent Doyen, Mickael Randour, and Jean-François Raskin.
Looking at mean-payoff and total-payoff through windows.
Inf. Comput., 242:25–52, 2015.

Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar, and Mariëlle Stoelinga.
The element of surprise in timed games.
In Roberto M. Amadio and Denis Lugiez, editors, CONCUR 2003 - Concurrency Theory, 14th International Conference, Marseille, France, September 3-5, 2003, Proceedings, volume 2761 of Lecture Notes in Computer Science, pages 142–156. Springer, 2003.

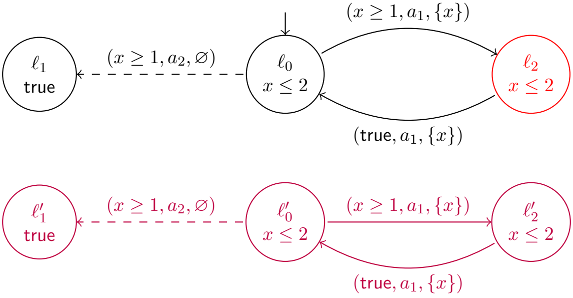
Complexity lower bound

- We can show EXPTIME-hardness by reducing the realizability problem for timed safety games to the realizability problem for timed window parity games.
- Due to time-convergence and divergence, we cannot use a sink state with an odd priority.



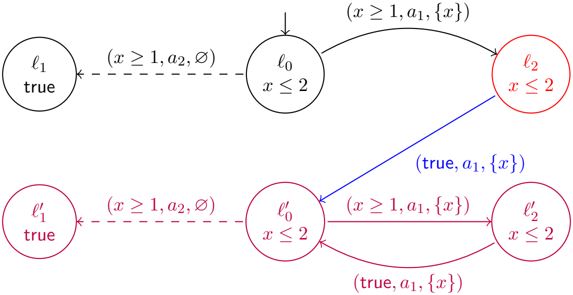
Complexity lower bound

- We can show EXPTIME-hardness by reducing the realizability problem for timed safety games to the realizability problem for timed window parity games.
- Due to time-convergence and divergence, we cannot use a sink state with an odd priority.



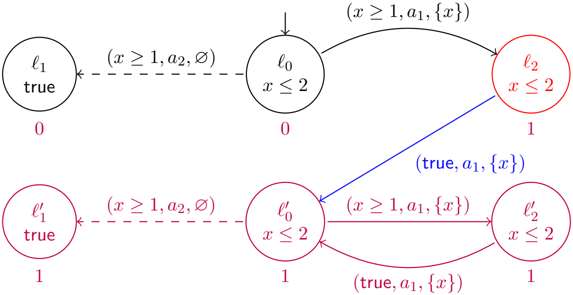
Complexity lower bound

- We can show EXPTIME-hardness by reducing the realizability problem for timed safety games to the realizability problem for timed window parity games.
- Due to time-convergence and divergence, we cannot use a sink state with an odd priority.



Complexity lower bound

- We can show EXPTIME-hardness by reducing the realizability problem for timed safety games to the realizability problem for timed window parity games.
- Due to time-convergence and divergence, we cannot use a sink state with an odd priority.



What happens if we change the semantics of winning ?

Bonus

In these theorems, time-divergence plays an important role.

Theorem

For all time-divergent plays π, π satisfies $\mathsf{DTW}(\lambda)$ *(resp.* $\mathsf{TW}(\lambda)$*) if and only if* $\mathsf{Ex}(\pi)$ *satisfies* $\mathsf{Safe}(\mathsf{Bad})$ *(resp.* $\mathsf{coBüchi}(\mathsf{Bad})$*).*

Theorem

For all time-divergent initial plays π of an expanded timed game, π satisfies $\mathsf{Safe}(\mathsf{Bad})$ *(resp.* $\mathsf{coBüchi}(\mathsf{Bad})$*) if and only if* $\mathsf{Pr}(\pi)$ *satisfies* $\mathsf{DTW}(\lambda)$ *(resp.* $\mathsf{TW}(\lambda)$*).*

Thanks to time-divergence, not satisfying the good window objective is equivalent to witnessing a bad window.

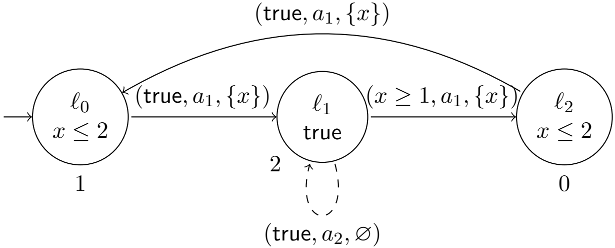
$\rightsquigarrow$ not true for all time-convergent plays !

What happens if we change the semantics of winning ?

Bonus

Consider the play π defined by:

$$(\ell_0, 0) \quad ((1, a_1), (2, \perp)) \quad (\ell_1, 0)$$
$$((1, a_1), (1/2, a_2)) \;\; (\ell_1, 1/2)$$
$$((1/2, a_1), (1/4, a_2))(\ell_1, 3/4) \ldots$$



$\hookrightarrow$ no good window but no bad window either for $\lambda = 2$

What happens if we change the semantics of winning ?

Bonus

- It is not sufficient to only consider a safety/Co-Büchi objective in the expanded game if we remove our time-divergence hypothesis.

Can we weaken our theorem's hypotheses ?

Erroneous theorem

For all ~~time-divergent~~ plays π, π satisfies $\mathsf{DTW}(\lambda)$ (resp. $\mathsf{TW}(\lambda)$) if and only if $\mathsf{Ex}(\pi)$ satisfies $\mathsf{Safe}(\mathsf{Bad})$ (resp. $\mathsf{coBüchi}(\mathsf{Bad})$).

$\rightsquigarrow$ problem with the direction " $\Longleftarrow$ "

What more should we ask of $\mathsf{Ex}(\pi)$?

What happens if we change the semantics of winning ?

Bonus

- Plays that are problematic are those that have windows that do not close, but are always of size strictly less than λ.

How do we know a window is closed in the expanded game ?

What happens if we change the semantics of winning ?
Bonus

- Plays that are problematic are those that have windows that do not close, but are always of size strictly less than λ.

  How do we know a window is closed in the expanded game ?

- A window is closed if and only if its smallest priority is even.

What happens if we change the semantics of winning ?

Bonus

- Plays that are problematic are those that have windows that do not close, but are always of size strictly less than λ.

How do we know a window is closed in the expanded game ?

- A window is closed if and only if its smallest priority is even.

Alternate theorem 1

For all ~~time-divergent~~ plays π, π satisfies $\mathsf{DTW}(\lambda)$ (resp. $\mathsf{TW}(\lambda)$) if and only if $\mathsf{Ex}(\pi)$ satisfies $\mathsf{Safe}(\mathsf{Bad})$ (resp. $\mathsf{coBüchi}(\mathsf{Bad})$) and $\mathsf{Büchi}(\mathsf{Even})$.

Alternate theorem 2

For all ~~time-divergent~~ initial plays π of an expanded timed game, π satisfies $\mathsf{Safe}(\mathsf{Bad}) \cap \mathsf{Büchi}(\mathsf{Even})$ (resp. $\mathsf{coBüchi}(\mathsf{Bad}) \cap \mathsf{Büchi}(\mathsf{Even})$) if and only if $\mathsf{Pr}(\pi)$ satisfies $\mathsf{DTW}(\lambda)$ (resp. $\mathsf{TW}(\lambda)$).