Qualitative Past Timeline-Based Games

Department of Mathematical, Computer and Physical Sciences, University of Udine

Renato Acampora renato.acampora@uniud.it
Luca Geatti luca.geatti@uniud.it
Nicola Gigante nicola.gigante@unibz.it
Angelo Montanari angelo.montanari@uniud.it

TIME 2023, ATHENS, 25 SEPTEMBER

Outline

- Timeline-based Planning (TP)
- Qualitative TP
- Restricting to Past

Overview

Timeline-Based Planning (TP)

Born in the context of space exploration [Mus92], Timeline-based Planning (TP) is an alternative to more common action-based approaches to planning:

- no clear separation between actions, states, and goals
- planning domain seen as a system composed of independent but interacting components
- behavior of such components, the timelines, governed by temporal constraints

[Mus92] - Frederking, Robert E., and Nicola Muscettola. "Temporal planning for transportation planning and scheduling." *ICRA*. 1992.

Timeline-Based Planning (TP)

Key features

- Modeling perspective: Timeline-based planning models problems in terms of components and timelines rather than states, actions, and goals. This focuses on temporal relationships rather than what actions achieve goals.

- Declarative nature: The modeling is more declarative, specifying what must/must not happen rather than how to achieve goals.

- Integrated temporal reasoning: Time is a first-class concept through timed tokens rather than an extra layer tacked onto actions. This fits domains with strong temporal constraints.

- Modular representation: Timelines of separate components can be modeled individually rather than a joint model. This supports distributed knowledge engineering.

Timeline-Based Games

Traditional approaches were focused on temporal uncertainty [COU16] (when things will happen), leaving behind general nondeterminism (what will happen).

Timeline-based games were formulated to fill this gap [Gig+20]:

- game theoretic approach to timeline-based planning with uncertainty
- uniformly handles general nondeterminism and temporal uncertainty

[COU16] - Cialdea Mayer, Marta, Andrea Orlandini, and Alessandro Umbrico. "Planning and execution with flexible timelines: a formal account." Acta Informatica 53.6-8 (2016): 649-680.
[Gig+20] - Gigante, Nicola, et al. "On timeline-based games and their complexity." Theoretical Computer Science 815 (2020): 247-269.

Timeline-Based Games

Controller synthesis

Finding whether a winning strategy exists for timeline-based games is known to be 2EXPTIME-complete [Gig+20].

We proposed an algorithm for synthesizing a controller from a timeline-based game specification, which involves constructing a Deterministic Finite Automaton (DFA) and playing simple reachability games [Aca+22].

[Gig+20] - Gigante, Nicola, et al. "On timeline-based games and their complexity." Theoretical Computer Science 815 (2020): 247-269.
[Aca+22] - Acampora, Renato, et al. "Controller Synthesis for Timeline-based Games." EPTCS 370: 131.

Controller Synthesis

- Representation of timelines as event sequences: Use event sequences to represent the evolution of state variables over time, rather than reasoning directly about timelines.
- Matching structures: To track rule satisfaction, introduce matching structures which combine a Difference Bound Matrix (DBM) [Dil90] and matched/unmatched tokens. The DBM encodes temporal constraints.
- Automaton construction: Construct a deterministic automaton (DFA) that reads an event sequence and tracks matching structure evolution. States are sets of matching structures.
- Arena construction: The DFA is transformed into an arena where moves encode the game rounds rather than raw events.
- Controller synthesis: A reachability game is solved on the arena. The attractor computation yields the winning region and a positional winning strategy.
- Complexity: The matching structures cause a double exponential blowup, as all combinations must be considered. This limits applicability to very large problems.
- Next steps: Simpler fragments, like restricting rules, could gain tractability while retaining practical expressiveness.

Motivation

Motivation

The high complexity of timeline-based games

- We aim to explore simpler yet expressive fragments
  - Leveraging results about past operators in temporal logic specification to make the synthesis problem exponentially more efficient [Art+23, DeG+20].
  - We plan to apply these findings to timeline-based games by introducing restrictions on synchronization rules to constrain the behavior of the system.

[DeG+20] - De Giacomo, Giuseppe, et al. "Pure-past linear temporal and dynamic logic on finite traces." IJCAI. 2020.
[Art+23] - Artale, Alessandro, et al. "Complexity of safety and cosafety fragments of linear temporal logic." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 37. No. 5. 2023.

Timeline-Based Planning (TP)

Timeline-Based Planning (TP)

State variables

$$x = (V_x, D_x, T_x, \gamma_x)$$

- V_x : is the finite domain of the state variable x.
- $D_x : V_x \to Intv$ is the duration function where $Intv$ is the set of intervals of $\mathbb{R}_+$ with endpoints in $\mathbb{N} \cup \{\infty\}$.
- $T_x : V_x \to 2^{V_x}$ is the value transition function.
- $\gamma_x : V_x \to \{\mathsf{c}, \mathsf{u}\}$ is the controllability function

Timeline-Based Planning (TP)

Definitions: 1

- **Definition** (Token)**.** A token for x is a tuple $\tau = (x, v, d)$, where x is a state variable, $v \in V_x$ is the value held by the variable, and $d \in \mathbb{N}^+$ is the duration of the token.
- **Definition** (Timeline)**.** A timeline for a state variable $x = (V_x, D_x, T_x, \gamma_x)$ is a finite sequence $\pi = \langle \tau_1, \tau_2, \ldots \tau_k \rangle$ of tokens for x.
- **Definition** (Multi-timeline)**.** Given a finite set SV of state variables, a multi-timeline of SV is a mapping Π assigning to each state variable $x \in$ SV a timeline for x.
- **Definition** (Atom)**.** An atom ρ is either a clause of the form $a_1 \leq_I^{t_1,t_2} a_2$ (interval atom), or of the forms $a_1 \leq_I^{e_1,e_2} n$ or $n \leq_I^{e_1,e_2} a_1$ (time-point atom), where $a_1, a_2 \in \Sigma$, $I \in Intv$, $n \in \mathbb{N}$, and $e_1, e_2 \in \{s, e\}$.

Timeline-Based Planning (TP)

Definitions: 2

- **Definition** (Existential statement)**.** An existential statement $\mathscr{E}$ for a finite set SV of state variables is a statement of the form

$$\mathscr{E} := \exists a_1[x_1 = v_1] \cdots \exists a_n[x_k = v_n] \,.\, \mathscr{C},$$

where $\mathscr{C}$ is a conjunction of atoms, $a_i \in \Sigma, x_i \in \mathsf{SV}, v_i \in V_{x_i}$ for $1 \leq i \leq n$.

- **Definition** (Synchronisation rule)**.** A synchronisation rule $\mathscr{R}$ for a finite set SV of state variables is a rule of one of the forms:

$$a_0[x_0 = v_0] \rightarrow \mathscr{E}_1 \vee \mathscr{E}_2 \vee \ldots \vee \mathscr{E}_k \text{ (trigger rule)}$$
$$\top \rightarrow \mathscr{E}_1 \vee \mathscr{E}_2 \vee \ldots \vee \mathscr{E}_k \text{ (trigger-less rule)}$$

where $a_0 \in \Sigma, x_0 \in \mathsf{SV}, v_0 \in V_{x_0},$ and $\mathscr{E}_1, \ldots, \mathscr{E}_k$ are *existential statements*.

Timeline-Based Planning (TP)

Definitions: 3

- **Definition** (Semantics of synchronization rules)**.** Let Π be a multi-timeline of a set SV of state variables.
  - Given a trigger-less rule $\mathscr{R}$ of SV, Π satisfies $\mathscr{R}$ if Π satisfies some existential statement of $\mathscr{R}$.
  - Given a trigger rule $\mathscr{R}$ of SV with trigger $a_0[x_0 = v_0]$, Π satisfies $\mathscr{R}$ if, for every position i of the timeline $\pi = \Pi(x_0)$ for x_0 such that $\pi(i) = (v_0, d)$, there are an existential statement $\mathscr{E}$ of $\mathscr{R}$ and a $\mathscr{N}$-assignment λ_Π for Π consistent with $\mathscr{E}$ such that $\lambda_\Pi(a_0) = (\pi, i)$ and λ_Π satisfies all the atoms of $\mathscr{E}$.

Qualitative TP

Qualitative TP

A starting point for efficient synthesis

- We choose to start from the qualitative fragment of timeline-based planning, which considers only qualitative (ordering) features of timelines [DeM+20].
- Nonetheless we expect to be able to extend the past restriction also to the quantitative case in the near future following the work presented in [Cim+21].

[DeM+20] - Della Monica, Dario, et al. "Complexity of qualitative timeline-based planning." 27th International Symposium on Temporal Representation and Reasoning (TIME 2020). Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
[Cim+21] - Cimatti, Alessandro, et al. "Extended bounded response LTL: a new safety fragment for efficient reactive synthesis." Formal Methods in System Design (2021): 1-49.

Qualitative TP

Definitions: 4

- **Definition** (Qualitative atom)**.** An atom $a_1 \leq^{t_1,t_2}_{[l,u]} a_2$ with $l = 0$ and $u = +\infty$ is said to be qualitative.
- **Definition** (Qualitative TP domain)**.** A TP domain $P = (\mathsf{SV}, S)$ is specified by a finite set SV of state variables with duration $D_x(v) = (1, +\infty)$ for each $x \in \mathsf{SV}$ and $v \in V_x$, and a finite set S of synchronization rules which make use only of qualitative atoms.
- **Definition** (Plan). A plan for $P = (\mathsf{SV}, S)$ is a multi-timeline of SV satisfying all rules in S.
- **Definition** (Qualitative TP problem)**.** Given a qualitative TP domain $P = (\mathsf{SV}, S)$ is there a plan for P?

Qualitative TP

An example

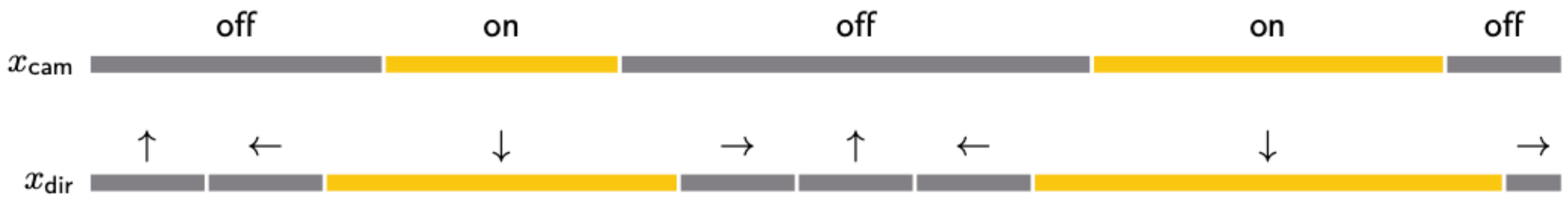
- The problem involves two state variables, x_{cam} and x_{dir}, with possible values $V_{x_{cam}} = \{\text{on}, \text{off}\}$ and $V_{x_{dir}} = \{ \uparrow, \leftarrow, \downarrow, \rightarrow \}$
- The transition function $T_{x_{dir}}$ allows the camera to move counterclockwise or stay still: $T_{x_{dir}}(\leftarrow) = \{ \leftarrow, \downarrow \}$
- The rules are built on a set $\mathcal{N} = \{a, b, \ldots\}$ of token names
- The first rule will require the camera to point down when it is switched on (e.g., to point toward the ground from an airplane)
- The system's goal will be to perform two shots with the camera switching off between them to cool down

Qualitative Timeline-Based Planning

An example

$$a[x_{cam} = \text{on}] \rightarrow \exists b[x_{dir} = \downarrow] \,.\, a \text{ during } b$$

$$\top \rightarrow \exists a[x_{cam} = \text{on}] b[x_{cam} = \text{off}] c[x_{cam} = \text{on}] \,.\, a \text{ meets } b \wedge end(a) \leq start(c)$$



The timelines involving state variables x_{cam} and x_{dir} representing the camera's on/off state and its pointing direction, respectively [DeM+20].

Qualitative TP

Automata based solution

- Synchronization rules are represented as preorders called blueprints, compactly representing ways to match a rule disjunct.
- Viewpoints are introduced to dynamically track how a blueprint is being matched to a plan. They contain a blueprint and frontier points separating matched from unmatched parts (Section 2.2).
- An automaton construction is defined where states contain compatible sets of viewpoints satisfying a symmetry-breaking condition (Section 3.3).
- Plans and words are shown to be in one-to-one correspondence by associating blueprint instantiations with triggers in a plan (Section 3.1).
- The automaton is shown to accept a word iff the corresponding plan solves the problem, using the covers and evolution of viewpoints (Section 3.4).
- This proves the problem can be solved in polynomial space by checking the automaton for emptiness (Section 3).
- PSPACE-hardness is proven by a reduction from intersection of finite automata (Section 4).

Restricting to Past

Restricting to Past

Desiderata

- The size of the deterministic automaton being exponential in the size of the planning problem
- A deterministic automaton which will serve as the game arena for achieving synthesis in exponential time.

Restricting to Past

Past synchronisation rules

- **Definition** (Past semantics of trigger rules)**.** A multi-timeline Π of SV satisfies a trigger rule

$$\mathscr{R} \equiv a_0[x_0 = v_0] \rightarrow \mathscr{E}_1 \vee \mathscr{E}_2 \vee \ldots \vee \mathscr{E}_k$$

under the past semantics if Π satisfies the trigger rule obtained from $\mathscr{R}$ by replacing each existential statement $\mathscr{E}_i \equiv \exists a_1[x_1 = v_1] \cdots \exists a_k[x_k = v_k] \, . \, \mathscr{C}$ by

$$\mathscr{E}'_i \equiv \exists a_1[x_1 = v_1] \cdots \exists a_n[x_n = v_n] \, . \left(\mathscr{C} \wedge \bigwedge_{i=1}^{n} a_i \leq^{e,s} a_0 \right)$$

Restricting to Past

Qualitative past TP

- **Definition** (Past Plan)**.** A past plan for $P = (S\mathsf{V}, S)$ is a multi-timeline of $S\mathsf{V}$ satisfying all rules in S under the past semantics of all trigger rules.
- **Definition** (Qualitative past TP problem)**.** Given a qualitative TP domain $P = (S\mathsf{V}, S)$, is there a *past* plan for P?

Restricting to Past

Automaton

- We build an automaton working on the partial orders defined by each synchronization rule (each existential statement to be precise).
- In contrast to the fragment accepting also the occurrence of future actions, with the past semantics we can read words (the past plans) deterministically, and it suffices to use downward closures as states of the automaton.
- The size of the automaton is exponential with regards to the size of the problem

Restricting to Past

Downward closures

- **Definition** (Downward closure)**.** Given an element x of a partially ordered set $(X, \leq)$, the downward closure of x, denoted by $\downarrow x$, is defined as

$$\downarrow x = \{l \in X : l \leq x\}.$$

Given a subset $A \subseteq X$, the downward closure of A, denoted by $A^{\downarrow X}$, is defined as

$$A^{\downarrow X} = \bigcup_{a \in A} \downarrow a.$$

Restricting to Past
The automaton

- We can build an automaton for the past planning problem by taking the cartesian product of all the automata resulting from each past synchronization rule.
- The automaton has the following components:
  - A set of states Q represented by all downward closures over the domain of existential statements,
  - The alphabet Σ is the superset of all possible actions
  - The transition function, which we informally define as follows:
    - If adding to the current state results in a downward closure, then we are okay and we stay in the same state.
    - Otherwise, we ignore the symbol until we either encounter the trigger token (in which case we transition to the sink state) or we encounter a symbol that we cannot ignore (the end of a previously started token). In this case, we go back to the state that has its downward closure resulting from getting rid of the upward closure of the element we are forced to match.
  - The initial state is the empty set
  - All states are final.

Future Directions

Future Directions

- Formalisation of Qualitative TP Games
- To adapt the work over the quantitative framework of TP, formalizing the fragment, analyzing its complexity, and developing effective algorithms.
- Explore a "cosafety fragment" of Timed Propositional Temporal Logic with Past ($TPTL_b + P$) that captures past timeline-based problems [DeM+17].

[DeM+17] - Della Monica, Dario, et al. "Bounded timed propositional temporal logic with past captures timeline-based planning with bounded constraints." IJCAI. International Joint Conferences on Artificial Intelligence, 2017.

References

- [Mus92] - Frederking, Robert E., and Nicola Muscettola. "Temporal planning for transportation planning and scheduling." *ICRA*. 1992.
- [COU16] - Cialdea Mayer, Marta, Andrea Orlandini, and Alessandro Umbrico. "Planning and execution with flexible timelines: a formal account." *Acta Informatica* 53.6-8 (2016): 649-680.
- [Gig+20] - Gigante, Nicola, et al. "On timeline-based games and their complexity." *Theoretical Computer Science* 815 (2020): 247-269.
- [Aca+22] - Acampora, Renato, et al. "Controller Synthesis for Timeline-based Games." *EPTCS 370*: 131.
- [Art+23] - Artale, Alessandro, et al. "Complexity of safety and cosafety fragments of linear temporal logic." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. No. 5. 2023.
- [Cim+21] - Cimatti, Alessandro, et al. "Extended bounded response LTL: a new safety fragment for efficient reactive synthesis." Formal Methods in System Design (2021): 1-49.

References

- [DeG+20] - De Giacomo, Giuseppe, et al. "Pure-past linear temporal and dynamic logic on finite traces." *IJCAI*. 2020.
- [DeM+20] - Della Monica, Dario, et al. "Complexity of qualitative timeline-based planning." *27th International Symposium on Temporal Representation and Reasoning (TIME 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
- [DeM+17] - Della Monica, Dario, et al. "Bounded timed propositional temporal logic with past captures timeline-based planning with bounded constraints." *IJCAI*. International Joint Conferences on Artificial Intelligence, 2017.
- [McM93] - McMillan, Kenneth L. *Symbolic model checking*. Springer US, 1993.
- [Bur+92] - Burch, Jerry R., et al. "Symbolic model checking: 1020 states and beyond." Information and computation 98.2 (1992): 142-170.
- [Tab19] - Tabajara, Lucas M., and Moshe Y. Vardi. "Partitioning techniques in LTLf synthesis." *2019 International Joint Conference on Artificial Intelligence*. 2019.

References

- [Dil90] - Dill, David L. "Timing assumptions and verification of finite-state concurrent systems." Automatic Verification Methods for Finite State Systems: International Workshop, Grenoble, France June 12–14, 1989 Proceedings 1. Springer Berlin Heidelberg, 1990.