Planning for Pure-Past Linear Temporal Goals

Francesco Fuggitti

fuggitti@diag.uniroma1.it

Joint work with Luigi Bonassi, Giuseppe De Giacomo, Marco Favorito, Alfonso Gerevini and Enrico Scala

AAAI 2023 Spring Symposium

SAPIENZA UNIVERSITÀ DI ROMA

YORK UNIVERSITÉ UNIVERSITY

Planning for Temporally Extended Goals

- Capture a richer class of plans using temporal logics
  - Deterministic planning [Bacchus et al. 1996; 1997; DeGiacomo&Vardi 1999; Bacchus&Kabanza 2000; …]
  - Planning via Model Checking [Cimatti et al. 1997; 1998; Giunchiglia&Traverso 1999; …]

- Recently, growing interest in the use of the finite-trace variant of LTL
  - Deterministic planning [Baier&McIlraith 2006; Torres&Baier 2015; …]
  - Nondeterministic domain models (FOND) [Camacho et al. 2017; DeGiacomo&Rubin 2018; …]

	Reachability Goals	Temporally Extended Goals (LTLf/LDLf)
Deterministic Planning	PSPACE-complete	PSPACE-complete
Nondeterministic Planning	EXPTIME-complete	2EXPTIME-complete

Pure-Past Linear Temporal Logic (PPLTL)

- Looks at the trace backward, and evaluates formulas on the last instant of the trace (i.e., the current instant)
- Past temporal operators only: (*Y)esterday, (S)ince, (O)nce in the past, (H)istorically*

Computational properties:

- As expressive as LTLf, but translating one into the other is prohibitive (3EXPTIME) [DeGiacomo et al. 2020]
- PPLTL to DFA is worst-case *single* exponential (vs. *double* exponential for LTLf to DFA) [Chandra et al. 1981; DeGiacomo et al. 2020]

PPLTL in Planning

- Little attention to AI planning, but commonly employed in other areas of AI
  - non-Markovian rewards in MDPs [Bacchus et al. 1996]
  - non-Markovian models [Gabaldon2011]
  - norms in multi-agent systems [Fisher&Wooldridge2005; Knobbout et al. 2016; Alechina et al. 2018]
- Actually, many interesting properties expressed in LTLf are *polynomially* related (in their size) to their *semantic* equivalent PPLTL (and vice versa)

DECLARE Template	Equivalent PPLTL Formula	Equivalent LTL$_f$ Formula
init(a)	$\mathsf{O}(a \land \neg\mathsf{Y}(true))$	a
existence(a)	$\mathsf{O}(a)$	$\mathsf{F}(a)$
absence(a)	$\neg\mathsf{O}(a)$	$\neg\mathsf{F}(a)$
absence2(a)	$\mathsf{H}(a \rightarrow \mathsf{WYH}(\neg a))$	$\neg(\mathsf{F}a \land \mathsf{XF}(a))$
choice(a, b)	$\mathsf{O}(a) \lor \mathsf{O}(b)$	$\mathsf{F}(a) \lor \mathsf{F}(b)$
exclusive-choice(a, b)	$(\mathsf{O}(a) \lor \mathsf{O}(b)) \land \neg(\mathsf{O}(a) \land \mathsf{O}(b))$	$(\mathsf{F}(a) \lor \mathsf{F}(b)) \land \neg(\mathsf{F}(a) \land \mathsf{F}(b))$
co-existence(a, b)	$\mathsf{H}(\neg a) \leftrightarrow \mathsf{H}(\neg b)$	$\mathsf{F}(a) \leftrightarrow \mathsf{F}(b)$
responded-existence(a, b)	$\mathsf{O}(a) \rightarrow \mathsf{O}(b)$	$\mathsf{F}(a) \rightarrow \mathsf{F}(b)$
response(a, b)	$(\neg a\, \mathsf{S}\, b) \lor \mathsf{H}(\neg a)$	$\mathsf{G}(a \rightarrow \mathsf{F}(b))$
precedence(a, b)	$\mathsf{H}(b \rightarrow \mathsf{O}(a))$	$(\neg b\, \mathsf{U}\, a) \lor \mathsf{G}(\neg b)$
succession(a, b)	response$(a, b) \land$ precedence(a, b)	
chain-response(a, b)	$\mathsf{H}(\mathsf{Y}(a) \rightarrow b) \land \neg a$	$\mathsf{G}(a \rightarrow \mathsf{X}(b))$
chain-precedence(a, b)	$\mathsf{H}(b \rightarrow \mathsf{Y}(a))$	$\mathsf{G}(\mathsf{X}(b) \rightarrow a) \land \neg b$
chain-succession(a, b)	$(\mathsf{H}(\mathsf{Y}(a) \rightarrow b) \land \neg a) \land$ $\mathsf{H}(\mathsf{Y}(\neg a) \rightarrow \neg b)$	$\mathsf{G}(a \leftrightarrow \mathsf{X}(b))$
not-co-existence(a, b)	$\mathsf{O}(a) \rightarrow \neg\mathsf{O}(b)$	$\mathsf{F}(a) \rightarrow \neg\mathsf{F}(b)$
not-succession(a, b)	$\mathsf{H}(b \rightarrow \neg\mathsf{O}(a))$	$\mathsf{G}(a \rightarrow \neg\mathsf{F}(b))$
not-chain-succession(a, b)	$\mathsf{H}(b \rightarrow \neg\mathsf{Y}(a))$	$\mathsf{G}(a \rightarrow \neg\mathsf{X}(b))$

PDDL3 Operator	Equivalent PPLTL Formula	Equivalent LTL$_f$ Formula
(at-end θ)	θ	$\mathsf{F}(\theta \land \mathsf{end})$
(always θ)	$\mathsf{H}(\theta)$	$\mathsf{G}(\theta)$
(sometime θ)	$\mathsf{O}(\theta)$	$\mathsf{F}(\theta)$
(sometime-after θ_1 θ_2)	$(\neg\theta_1\, \mathsf{S}\, \theta_2) \lor \mathsf{H}(\neg\theta_1)$	$\mathsf{G}(\theta_1 \rightarrow \mathsf{F}(\theta_2))$
(sometime-before θ_1 θ_2)	$\mathsf{H}(\theta_1 \rightarrow \mathsf{Y}(\mathsf{O}(\theta_2)))$	$\theta_2\, \mathsf{R}\, \neg\theta_1$
(at-most-once θ)	$\mathsf{H}(\theta \rightarrow (\theta\, \mathsf{S}\, (\mathsf{H}(\neg\theta) \lor \mathsf{start})))$	$\mathsf{G}(\theta \rightarrow (\theta\, \mathsf{U}\, (\mathsf{G}(\neg\theta) \lor \mathsf{end})))$
(hold-during n_1 n_2 θ)	$\bigvee_{0 \leq i \leq n_1} (\theta \land \mathsf{Y}^i(\mathsf{start})) \lor$ $\bigwedge_{n_1 < i \leq n_2} \mathsf{H}(\theta \lor \mathsf{WY}^i(\mathsf{Y}(true)))$	$\bigvee_{0 \leq i \leq n_1} \mathsf{X}^i(\theta \land \mathsf{end}) \lor$ $\bigwedge_{n_1 < i \leq n_2} \mathsf{WX}^i(\theta)$
* (hold-after n θ)	$\bigvee_{0 \leq i \leq n} (\theta \land \mathsf{Y}^i(\mathsf{start})) \lor$ $\mathsf{O}(\theta \land \mathsf{Y}^{n+1}(\mathsf{O}(\mathsf{start})))$	$\bigvee_{0 \leq i \leq n} \mathsf{X}^i(\theta \land \mathsf{end}) \lor$ $\mathsf{X}^{n+1}(\mathsf{F}(\theta))$

Handling PPLTL Goals

Intuition: given the prefix of a trace, while LTLf has to consider all possible extensions, PPLTL can simply be evaluated on the prefix (i.e., the history produced so far)

How?
Exploit the “fixpoint characterization” of temporal formulas [Gabbay et al. 1980; Manna 1982; Barringer et al., 1989; Emerson 1990]

- $\mathsf{pnf}(p) = p$;
- $\mathsf{pnf}(\mathsf{Y}\phi) = \boxed{\mathsf{Y}\phi;}$
- $\mathsf{pnf}(\phi_1 \mathbin{\mathsf{S}} \phi_2) = \mathsf{pnf}(\phi_2) \lor (\mathsf{pnf}(\phi_1) \land \boxed{\mathsf{Y}(\phi_1 \mathbin{\mathsf{S}} \phi_2));}$
- $\mathsf{pnf}(\phi_1 \land \phi_2) = \mathsf{pnf}(\phi_1) \land \mathsf{pnf}(\phi_2)$;
- $\mathsf{pnf}(\neg\phi) = \neg\mathsf{pnf}(\phi)$.

To evaluate a PPLTL formula, we only need to keep track of the truth value of *some* of its subformulas!!!

Evaluating PPLTL Goals

Technique

- Collect these key subformulas as *propositions* in a set Σ_φ
- Define an interpretation function $\sigma: \Sigma_\varphi \rightarrow \{\top, \bot\}$ that tells which propositions are true at a given instant of time
- Given the propositional interpretation of the *current instant* s_i and truth value σ_i of propositions in Σ_φ, evaluate any PPLTL formulas at instant *i* through val() predicate recursively as follows:

- $\mathsf{val}(p, \sigma_i, s_i)$ *iff* $s_i \models p$;
- $\mathsf{val}(\mathsf{Y}\phi', \sigma_i, s_i)$ *iff* $\sigma_i \models \boxed{\text{“}\mathsf{Y}\phi'\text{”};}$
- $\mathsf{val}(\phi_1 \,\mathsf{S}\, \phi_2, \sigma_i, s_i)$ *iff* $\mathsf{val}(\phi_2, \sigma_i, s_i) \vee (\mathsf{val}(\phi_1, \sigma_i, s_i) \wedge \sigma_i \models \boxed{\text{“}\mathsf{Y}(\phi_1 \,\mathsf{S}\, \phi_2)\text{”});}$
- $\mathsf{val}(\phi_1 \wedge \phi_2, \sigma_i, s_i)$ *iff* $\mathsf{val}(\phi_1, \sigma_i, s_i) \wedge \mathsf{val}(\phi_2, \sigma_i, s_i)$;
- $\mathsf{val}(\neg\phi', \sigma_i, s_i)$ *iff* $\neg\mathsf{val}(\phi', \sigma_i, s_i)$.

Theorem

Given $<\sigma_0,\ldots,\sigma_n>$, a trace $<s_0,\ldots,s_n>$ satisfies a PPLTL formula φ *if and only if* $\mathsf{val}(\varphi, \sigma_n, s_n)$

Planning for PPLTL Goals

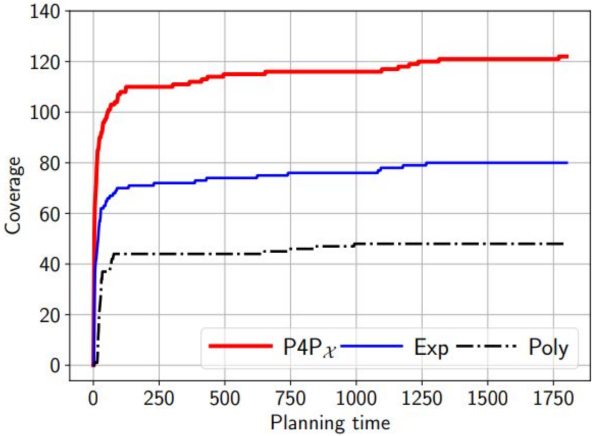
- Introduce only *few* new fluents, at most **linear** in the size of the PPLTL goal, i.e. minimal overhead
- No spurious additional actions
- Sidestep altogether the standard automata construction

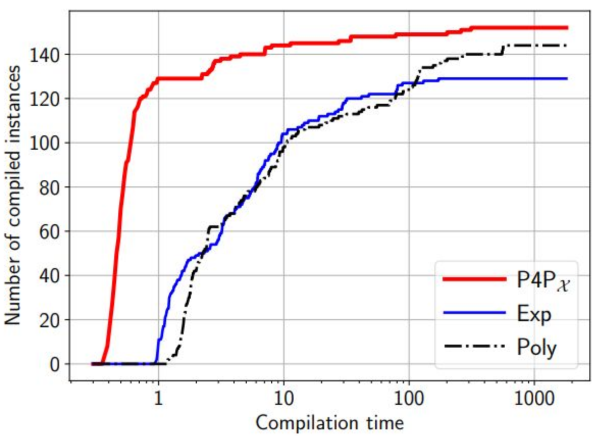
Components	Encoding
Fluents $\mathcal{F}'$	$\mathcal{F}' := \mathcal{F} \cup \{\text{"}\mathsf{Y}\phi\text{"} \mid \text{"}\mathsf{Y}\phi\text{"} \in \Sigma_\varphi\}$
Derived Predicates $\mathcal{F}'_{der}$	$\mathcal{F}'_{der} := \mathcal{F}_{der} \cup \{\mathsf{val}_\phi \mid \phi \in \mathsf{sub}(\varphi)\}$
Axioms $\mathcal{X}'$	$\mathcal{X}' := \mathcal{X} \cup \{x_\phi \mid \phi \in \mathsf{sub}(\varphi)\}$ where x_ϕ is $\begin{cases} \mathsf{val}_p \leftarrow p & (\phi = p) \\ \mathsf{val}_{\mathsf{Y}\phi'} \leftarrow \text{"}\mathsf{Y}\phi'\text{"} & (\phi = \mathsf{Y}\phi') \\ \mathsf{val}_{\phi_1 \mathsf{S} \phi_2} \leftarrow (\mathsf{val}_{\phi_2} \vee (\mathsf{val}_{\phi_1} \wedge \text{"}\mathsf{Y}(\phi_1 \mathsf{S} \phi_2)\text{"})) & (\phi = \phi_1 \mathsf{S} \phi_2) \\ \mathsf{val}_{\phi_1 \wedge \phi_2} \leftarrow (\mathsf{val}_{\phi_1} \wedge \mathsf{val}_{\phi_2}) & (\phi = \phi_1 \wedge \phi_2) \\ \mathsf{val}_{\neg\phi'} \leftarrow \neg \mathsf{val}_{\phi'} & (\phi = \neg\phi') \end{cases}$
Action Labels A	$A := A$, i.e., unchanged
Preconditions pre	$pre(a) := pre(a)$ for every $a \in A$, i.e., unchanged
Effects eff'	$\mathit{eff}'(a) := \{\mathsf{eff}_i \cup \mathsf{eff}_{\mathsf{val}} \mid \mathsf{eff}_i \in \mathit{eff}(a)\}$, where $\mathsf{eff}_{\mathsf{val}} = \{\mathsf{val}_\phi \rhd \{\text{"}\mathsf{Y}\phi\text{"}\}, \neg\mathsf{val}_\phi \rhd \{\neg\text{"}\mathsf{Y}\phi\text{"}\} \mid \text{"}\mathsf{Y}\phi\text{"} \in \Sigma_\varphi\}$
Initial State s'_0	$s'_0 := \sigma_0 \cup s_0$
Goal G'	$G' := \mathsf{val}_\varphi$

Sound and complete approach to symbolically encode PPLTL temporally extended goal formulas in planning domains that is *linear* in both the size of the domain specification and the size of the PPLTL goal

Results for Deterministic Planning

- Introduce the Plan4Past[1] system
- Compare Plan4Past against state-of-the-art techniques for LTLf, Exp [Baier&McIlraith 2006] and Poly [Torres&Baier 2015], on a set of equivalent (semantic- and size-wise) LTLf/PPLTL formulas
- IPC domains: BLOCKS, ELEVATOR, OPENSTACKS, ROVERS





[1] https://github.com/whitemech/Plan4Past

Summary and Future Work

- How to efficiently handle and evaluate PPLTL formulas
- Sound and complete approach to solve planning for PPLTL goals that is optimal wrt theoretical complexity with a clear advantage in practice
- To appear at ICAPS23: “Planning for Temporally Extended Goals in Pure-Past Linear Temporal Logic”

Future Work:

- Study nondeterministic planning
- Developing PPLTL-aware heuristics that exploit the structure of the formula
- Incorporate PPLTL patterns into PDDL, giving rise to PDDL4.0