



Temporally extended goal recognition in fully observable non-deterministic domain models

Temporally extended goal recognition in FOND planning

Ramon Fraga Pereira¹ · Francesco Fuggitti^{2,3,4} · Felipe Meneguzzi⁵ · Giuseppe De Giacomo^{3,6}

Accepted: 5 October 2023
© The Author(s) 2023

Abstract

Goal Recognition is the task of discerning the intended goal that an agent aims to achieve, given a set of goal hypotheses, a domain model, and a sequence of observations (i.e., a sample of the plan executed in the environment). Existing approaches assume that goal hypotheses comprise a single conjunctive formula over a single final state and that the environment dynamics are deterministic, preventing the recognition of temporally extended goals in more complex settings. In this paper, we expand goal recognition to *temporally extended goals* in *Fully Observable Non-Deterministic* (FOND) planning domain models, focusing on goals on finite traces expressed in *Linear Temporal Logic* (LTL_f) and *Pure-Past Linear Temporal Logic* (PPLTL). We develop the first approach capable of recognizing goals in such settings and evaluate it using different LTL_f and PPLTL goals over six FOND planning domain models. Empirical results show that our approach is accurate in recognizing temporally extended goals in different recognition settings.

Keywords Automated planning · Goal recognition · Non-deterministic planning · Linear temporal logic.

1 Introduction

Goal Recognition is the task of recognizing the intentions of autonomous agents or humans by observing their interactions in an environment. Existing work on goal and plan recognition addresses this task over several different types of domain settings, such as plan-libraries [4], plan tree grammars [19], classical planning domain models [31, 34, 35, 37], stochastic environments [36], continuous domain models [22], incomplete discrete domain models [29], and approximate control models [30]. Despite the ample literature and recent advances, most existing approaches to *Goal Recognition as Planning* cannot recognize *temporally extended goals*, i.e., goals formalized in terms of time, e.g., the exact order that a set of facts of a goal must be achieved in a plan. Recently, [1] propose a general formulation of a temporal inference problem in deterministic planning settings. However, most of these approaches also assume that the observed actions' out-

comes are deterministic and do not deal with unpredictable, possibly adversarial, environmental conditions.

Research on planning for *temporally extended goals* in *deterministic* and *non-deterministic* domains has increased over the years, starting with the pioneering work on planning for temporally extended goals [5] and on planning via model checking [12]. This continued with the work on integrating LTL goals into planning tools [27, 28], and, most recently, [7], introducing a novel optimal encoding of Pure-Past Linear Temporal Logic goals into for *Classical Planning*. Other existing works relate *program synthesis* [33] with planning in non-deterministic domains for temporal specifications, recently focusing on the *finite trace* variants of LTL [2, 9, 10, 14–16].

In this paper, we introduce the task of goal recognition in *discrete domains* that are *fully observable*, and the outcomes of actions and observations are *non-deterministic*, possibly adversarial, i.e., *Fully Observable Non-Deterministic* (FOND), allowing the formalization of *temporally extended goals* using two types of temporal logic on finite traces: *Linear-time Temporal Logic* (LTL_f) and *Pure-Past Linear-time Temporal Logic* (PPLTL) [17].

The main contribution of this paper is three-fold. First, based on the definition of *Plan Recognition as Planning*

✉ Ramon Fraga Pereira
ramon.fragapereira@manchester.ac.uk

Extended author information available on the last page of the article

introduced in [34], we formalize *the problem of recognizing temporally extended goals* (expressed in LTL_f or PPLTL) in FOND planning domains, handling both stochastic (i.e., strong-cyclic plans) and adversarial (i.e., strong plans) environments [2]. Second, we extend the probabilistic framework for goal recognition proposed in [35], and develop a novel *probabilistic approach* that reasons over executions of policies and returns a posterior probability distribution for the goal hypotheses. Third, we develop a *compilation approach* that generates an augmented FOND planning problem by compiling temporally extended goals together with the original planning problem. This compilation allows us to use any off-the-shelf FOND planner to perform the recognition task in FOND planning models with temporally extended goals.

This work focuses on FOND domains with stochastic non-determinism, and conduct an extensive set of experiments with different complex problems. We empirically evaluate our approach using different LTL_f and PPLTL goals over six FOND planning domain models, including a real-world non-deterministic domain model [26], and our experiments show that our approach is accurate to recognize temporally extended goals in different two recognition settings: *offline recognition*, in which the recognition task is performed in “one-shot”, and the observations are given at once and may contain missing information; and *online recognition*, in which the observations are received incrementally, and the recognition task is performed gradually.

2 Preliminaries

This section briefly recalls the syntax and semantics of *Linear-time Temporal Logics* on finite traces (LTL_f and PPLTL) and revises the concept and terminology of FOND planning.

2.1 LTL_f and PPLTL

Linear Temporal Logic on finite traces (LTL_f) is a variant of LTL introduced in [32] interpreted over *finite traces*. Given a set of atomic propositions AP , the syntax of LTL_f formulas φ is defined as follows:

$$\varphi ::= a \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc\varphi \mid \varphi\mathcal{U}\varphi$$

where a denotes an atomic proposition in AP , \bigcirc is the *next* operator, and \mathcal{U} is the *until* operator. Apart from the Boolean connectives, we use the following abbreviations: *eventually* as $\Diamond\varphi \doteq \text{true}\mathcal{U}\varphi$; *always* as $\Box\varphi \doteq \neg\Diamond\neg\varphi$; *weak next* $\bullet\varphi \doteq \neg\bigcirc\neg\varphi$. A trace $\tau = \tau_0\tau_1\cdots$ is a sequence of propositional interpretations, where $\tau_m \in 2^{AP}$ ($m \geq 0$) is the m -th interpretation of τ , and $|\tau|$ is the length of τ . We denote a finite trace formally as $\tau \in (2^{AP})^*$. Given a finite

trace τ and an LTL_f formula φ , we inductively define when φ holds in τ at position i ($0 \leq i < |\tau|$), written $\tau, i \models \varphi$ as follows:

- $\tau, i \models a$ iff $a \in \tau_i$;
- $\tau, i \models \neg\varphi$ iff $\tau, i \not\models \varphi$;
- $\tau, i \models \varphi_1 \wedge \varphi_2$ iff $\tau, i \models \varphi_1$ and $\tau, i \models \varphi_2$;
- $\tau, i \models \bigcirc\varphi$ iff $i + 1 < |\tau|$ and $\tau, i + 1 \models \varphi$;
- $\tau, i \models \varphi_1\mathcal{U}\varphi_2$ iff there exists j such that $i \leq j < |\tau|$ and $\tau, j \models \varphi_2$, and for all k , $i \leq k < j$, we have $\tau, k \models \varphi_1$.

An LTL_f formula φ is *true* in τ , denoted by $\tau \models \varphi$, iff $\tau, 0 \models \varphi$. As advocated by [17], this paper also uses the *pure-past* version of LTL_f , here denoted as PPLTL, due to its compelling computational advantage compared to LTL_f when goal specifications are *naturally* expressed in a past fashion. PPLTL refers *only* to the past and has a natural interpretation on finite traces: formulas are satisfied if they hold in the current (i.e., last) position of the trace.

Given a set AP of propositional symbols, PPLTL formulas are defined by:

$$\varphi ::= a \mid \neg\varphi \mid \varphi \wedge \varphi \mid \ominus\varphi \mid \varphi\mathcal{S}\varphi$$

where $a \in AP$, \ominus is the *before* operator, and \mathcal{S} is the *since* operator. Similarly to LTL_f , common abbreviations are the *once* operator $\Diamond\varphi \doteq \text{true}\mathcal{S}\varphi$ and the *historically* operator $\Box\varphi \doteq \neg\Diamond\neg\varphi$. Given a finite trace τ and a PPLTL formula φ , we inductively define when φ holds in τ at position i ($0 \leq i < |\tau|$), written $\tau, i \models \varphi$ as follows. For atomic propositions and Boolean operators it is as for LTL_f . For past operators:

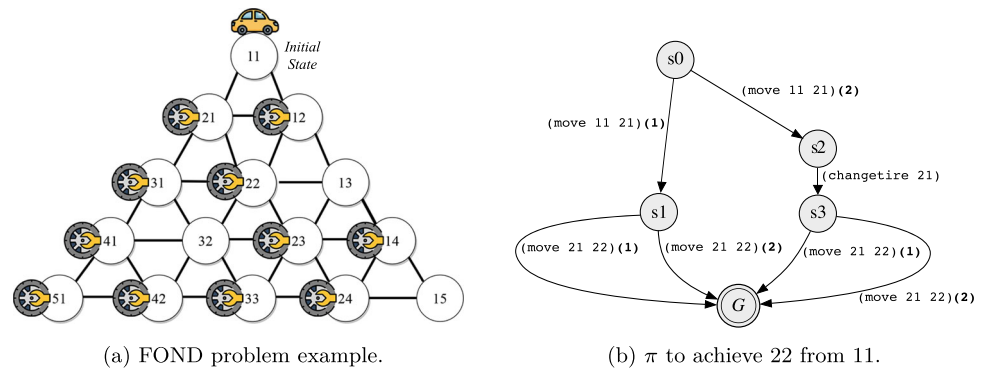
- $\tau, i \models \ominus\varphi$ iff $i - 1 \geq 0$ and $\tau, i - 1 \models \varphi$;
- $\tau, i \models \varphi_1\mathcal{S}\varphi_2$ iff there exists k such that $0 \leq k \leq i$ and $\tau, k \models \varphi_2$, and for all j , $k < j \leq i$, we have $\tau, j \models \varphi_1$.

A PPLTL formula φ is *true* in τ , denoted by $\tau \models \varphi$, if and only if $\tau, |\tau| - 1 \models \varphi$. A key property of temporal logics exploited in this work is that, for every LTL_f /PPLTL formula φ , there exists a *Deterministic Finite-state Automaton* (DFA) \mathcal{A}_φ accepting the traces τ satisfying φ [15, 17].

2.2 FOND planning

A *Fully Observable Non-deterministic Domain* planning model (FOND) is a tuple $\mathcal{D} = \langle 2^{\mathcal{F}}, A, \alpha, tr \rangle$ [18], where $2^{\mathcal{F}}$ is the set of possible states and \mathcal{F} is a set of fluents (atomic propositions); A is the set of actions; $\alpha(s) \subseteq A$ is the set

Fig. 1 TRIANGLE- TIREWORLD domain and policy



of applicable actions in a state s ; and $tr(s, a)$ is the non-empty set of successor states that follow action a in state s . A domain \mathcal{D} is assumed to be compactly represented (e.g., in PDDL [24]), hence its size is $|\mathcal{F}|$. Given the set of *literals* of \mathcal{F} as $Literals(\mathcal{F}) = \mathcal{F} \cup \{\neg f \mid f \in \mathcal{F}\}$, every action $a \in A$ is usually characterized by $\langle Pre_a, Eff_a \rangle$, where $Pre_a \subseteq Literals(\mathcal{F})$ is the action preconditions, and Eff_a is the action effects. An action a can be applied in a state s if the set of fluents in Pre_a holds true in s . The result of applying a in s is a successor state s' non-deterministically drawn from one of the Eff_a^i in $Eff_a = \{Eff_a^1, \dots, Eff_a^m\}$. In FOND planning, some actions have *uncertain outcomes*, such that they have *non-deterministic* effects (i.e., $|tr(s, a)| \geq 1$ in all states s in which a is applicable), and effects cannot be predicted in advance. PDDL expresses uncertain outcomes using the `oneof` [8] keyword, as widely used by several FOND planners [23, 25]. A FOND planning problem is formally defined as follows.

Definition 1 A **FOND planning problem** is a tuple $\mathcal{P} = \langle \mathcal{D}, s_0, G \rangle$, where \mathcal{D} is a FOND domain model, s_0 is an initial assignment to fluents in \mathcal{F} (i.e., initial state), and $G \subseteq \mathcal{F}$ is the goal.

Solutions to a FOND planning problem \mathcal{P} are *policies*. A policy is usually denoted as π , and formally defined as a partial function $\pi : 2^{\mathcal{F}} \rightarrow A$ mapping *non-goal* states into applicable actions that eventually reach a goal state complying with G from the initial state s_0 . We say a state s complies with G if $G \subseteq s$. A *policy* π for \mathcal{P} induces a set of possible *executions* $\vec{E} = \{\vec{e}_1, \vec{e}_2, \dots\}$, that are state trajectories, possibly finite (i.e., histories) (s_0, \dots, s_n) , where $s_{i+1} \in tr(s_i, a_i)$ and $a_i \in \alpha(s_i)$ for $i = 0, \dots, n-1$, or possibly infinite s_0, s_1, \dots , obtained by choosing some possible outcome of actions instructed by the policy. A policy π is a solution to \mathcal{P} if every execution is finite and satisfies the goal G in its last state, i.e., $s_n \models G$. In this case, π is *winning*. [13] define three solutions to FOND planning problems: *weak*, *strong* and *strong-cyclic* solutions, formally defined in Definitions 2, 4, and 3.

Definition 2 A **weak solution** is a policy that achieves a goal state complying with G from the initial state s_0 under at least one selection of action outcomes; namely, such solution will have some chance of achieving a goal state complying with G .

Definition 3 A **strong-cyclic solution** is a policy that guarantees to achieve a goal state complying with G from the initial state s_0 only under the assumption of fairness¹. However, this type of solution may revisit states, so the solution cannot guarantee to achieve G in a fixed number of steps.

Definition 4 A **strong solution** is a policy that is guaranteed to achieve a goal state complying with G from the initial state s_0 regardless of the environment's non-determinism. This type of solution guarantees the achievement of G in a finite number of steps while never visiting the same state twice.

This work focuses on *strong-cyclic solutions*, where the environment acts in an unknown but stochastic way. Nevertheless, our recognition approach applies to strong solutions as well, where the environment is purely adversarial (i.e., the environment may always choose effects against the agent).

Our running example comes from the well-known TRIANGLE- TIREWORLD FOND domain, where roads connect locations, and the agent can drive through them. The objective is to drive from one location to another. However, while driving between locations, a tire may go flat, and if there is a spare tire in the car's location, then the car can use it to fix the flat tire. Figure 1a illustrates a FOND planning problem for the TRIANGLE- TIREWORLD domain, where circles are locations, arrows represent roads, spare tires are depicted as tires, and the agent is depicted as a car. Figure 1b shows a policy π to achieve location 22. Note that, to move from location 11 to location 21, there are two arrows labeled with the action $(move\ 11\ 21) : (1)$ when moving does not cause the tire

¹ The fairness assumption defines that all action outcomes in a given state have a non-zero probability.

to go flat; (2) when moving causes the tire to go flat. The policy depicted in Fig. 1b guarantees the success of achieving location 22 despite the environment's non-determinism.

From *Classical Planning*, the cost for all *non-deterministic* instantiated actions $a \in A$ is 1. In this example, policy π , depicted in Fig. 1b, has two possible finite executions in the set of executions \bar{E} , namely $\bar{E} = \{\bar{e}_0, \bar{e}_1\}$, such as:

- \bar{e}_0 : [(move 11 21), (move 21 22)]; and
- \bar{e}_1 : [(move 11 21), (changetire 21), (move 21 22)].

3 FOND planning for LTL_f and PPLTL goals

We base our approach to goal recognition in FOND domains for *temporally extended goals* on FOND planning with LTL_f and PPLTL goals [9, 10, 14]. Definition 5 formalizes a FOND planning problem with LTL_f/PPLTL goals as follows.

Definition 5 A FOND planning problem with LTL_f/PPLTL goals is a tuple $\Gamma = \langle \mathcal{D}, s_0, \varphi \rangle$, where \mathcal{D} is a standard FOND domain model, s_0 is the initial state, and φ is a goal formula, formally represented either as an LTL_f or a PPLTL formula.

In FOND planning with temporally extended goals, a policy π is a partial function $\pi : (2^{\mathcal{F}})^+ \rightarrow A$ mapping *histories*, i.e., *states* into applicable actions. A policy π for Γ achieves a temporal formula φ if and only if the sequence of states generated by π , despite the non-determinism of the environment, is accepted by \mathcal{A}_φ .

Key to our recognition approach is encoding the temporal goal formula into an extended planning domain, expressed in PDDL, which can be later consumed by off-the-shelf FOND planners. Compiling planning for temporally extended goals into planning for standard *reachability* goals (i.e., final-state goals) has a long history in the AI *Planning* literature. In particular, [6] develops *deterministic* planning with special first-order quantified LTL goals on finite-state sequences. Their technique encodes a *Non-Deterministic Finite-state Automaton* (NFA), resulting from LTL formulas, into deterministic planning domains for which *Classical Planning* technology can be leveraged. Our parameterization of objects of interest is somehow similar to their approach. Starting from [6], always in the context of deterministic planning, [38] proposed a polynomial-time compilation of LTL goals on finite-state sequences into alternating automata, leaving non-deterministic choices to be decided at planning time. Finally, [9, 10] built upon [6] and [38], proposing a compilation in the context of FOND domain models that explicitly

computes the automaton representing the LTL_f temporal goal and encodes it into PDDL. However, this encoding introduces a lot of bookkeeping machinery due to the removal of any form of angelic non-determinism mismatching with the devilish non-determinism of PDDL for FOND.

Although inspired by such work, our approach differs in several technical details. We encode the DFA directly into a non-deterministic PDDL planning domain by taking advantage of the *parametric* nature of PDDL domains that are then instantiated into propositional problems when solving a specific task. Given a FOND planning problem Γ represented in PDDL, the transformation Γ works as follows. First, the highly-optimized MONA tool [20] transforms the temporally extended goal formula φ (formalized either in LTL_f or PPLTL) into its corresponding DFA \mathcal{A}_φ . Second, from \mathcal{A}_φ , we build a *parametric* DFA (PDFA), representing the lifted version of the DFA. Finally, the encoding of such a PDFA into PDDL yields an augmented FOND domain model Γ' . Thus, this process reduces FOND planning for LTL_f/PPLTL to a standard FOND planning problem solvable by any off-the-shelf FOND planner.

3.1 Translation to parametric DFA

The use of *parametric* DFAs is based on the following observations. In temporal logic formulas and, hence, in the corresponding DFAs, propositions are represented by domain fluents grounded on specific objects of interest. We can replace these propositions with predicates using object variables and then have a mapping function m^{obj} that maps such variables into the problem instance objects. This yields a lifted and *parametric* representation of the DFA, i.e., PDFA, which is merged with the domain. Here, the objective is to capture the entire dynamics of the DFA within the planning domain model itself. To do so, starting from the DFA we build a PDFA whose states and symbols are the lifted versions of the ones in the DFA. Formally, to construct a PDFA we use a mapping function m^{obj} , which maps the set of objects of interest present in the DFA to a set of *free* variables. Given the mapping function m^{obj} , Definition 6 formalizes a PDFA as follows.

Definition 6 Given a set of object symbols \mathcal{O} , and a set of free variables \mathcal{V} , we define a **mapping function** m that maps each object in \mathcal{O} with a free variable in \mathcal{V} .

Given a DFA and the objects of interest for Γ , we can construct a PDFA as follows:

Definition 7 A PDFA is a tuple $\mathcal{A}_\varphi^p = \langle \Sigma^p, Q^p, q_0^p, \delta^p, F^p \rangle$, where: $\Sigma^p = \{\sigma_0^p, \dots, \sigma_n^p\} = 2^{\mathcal{F}}$ is the alphabet of fluents;

Q^P is a nonempty set of parametric states; q_0^P is the parametric initial state; $\delta^P : Q^P \times \Sigma^P \rightarrow Q^P$ is the parametric transition function; $F^P \subseteq Q^P$ is the set of parametric final states. Σ^P , Q^P , q_0^P , δ^P and F^P can be obtained by applying m^{obj} to all the components of the corresponding DFA.

Example 1 Given the LTL_f formula “ $\Diamond(vAt\ 51)$ ”, the object of interest “51” is replaced by the object variable x (i.e., $m^{obj}(51) = x$), and the corresponding DFA and PDFA for this LTL_f formula are depicted in Fig. 2a and b.

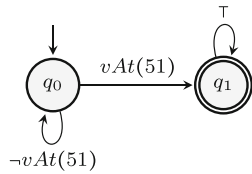
When the resulting new domain is instantiated, we implicitly get back the original DFA in the Cartesian product with the original instantiated domain. Note that this way of proceeding is similar to what is done in [6], where they handle LTL_f goals expressed in a special FOL syntax, with the resulting automata (non-deterministic Büchi automata) parameterized by the variables in the LTL_f formulas.

3.2 PDFA encoding in PDDL

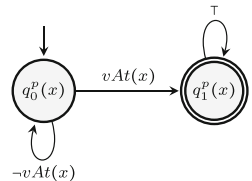
Once the PDFA has been computed, we encode its components within the planning problem Γ , specified in PDDL, thus, producing an augmented FOND planning problem $\Gamma' = \langle \mathcal{D}', s'_0, G' \rangle$, where $\mathcal{D}' = \langle 2^{\mathcal{F}'}, A', \alpha', tr' \rangle$ and G' is a propositional goal as in *Classical Planning*. Intuitively, additional parts of Γ' are used to synchronize the dynamics between the domain and the automaton sequentially. Specifically, Γ' is composed of the following components.

Fluents

\mathcal{F}' has the same fluents in \mathcal{F} plus fluents representing each state of the PDFA, and a fluent called `turnDomain`, which controls the alternation between domain's actions and the PDFA's synchronization action. Formally, $\mathcal{F}' = \mathcal{F} \cup \{q \mid q \in Q^P\} \cup \{\text{turnDomain}\}$.



(a) DFA for $\Diamond(vAt(51))$



(b) PDFA for $\Diamond(vAt(51))$

Fig. 2 DFA and PDFA for $\Diamond(vAt(51))$

Domain actions

Actions in A are modified by adding `turnDomain` in preconditions and the negated `turnDomain` in effects: $Pre'_a = Pre_a \cup \{\text{turnDomain}\}$ and $Eff'_a = Eff_a \cup \{\neg \text{turnDomain}\}$ for all $a \in A$.

Transition operator

The *transition* function δ^P of a PDFA is encoded as a new domain operator with conditional effects, called `trans`. Namely, $Pre_{\text{trans}} = \{\neg \text{turnDomain}\}$ and $Eff_{\text{trans}} = \{\text{turnDomain}\} \cup \{\text{when } (q^P, \sigma^P), \text{ then } \delta^P(q^P, \sigma^P) \cup \{\neg q \mid q \neq q^P, q \in Q^P\}\}$, for all $(q^P, \sigma^P) \in \delta^P$. To exemplify how the transition PDDL operator is obtained, Listing 1 reports the transition operator for the PDFA in Fig. 2.

```
(:action trans
:parameters (?x - location)
:precondition (not (turnDomain))
:effect (and
  (when (and (q0 ?x) (not (vAt ?x)))
    (and (q0 ?x) (not (q1 ?x)) (turnDomain))
  )
  (when (or (and (q0 ?x) (vAt ?x)) (q1 ?x))
    (and (q1 ?x) (not (q0 ?x))
      (turnDomain))))))
```

Listing 1 Transition PDDL operator for $\Diamond(vAt(x))$

Initial and goal states

The new initial condition is specified as $s'_0 = s_0 \cup \{q_0^P\} \cup \{\text{turnDomain}\}$. This comprises the initial condition of the previous domain D (s_0) plus the initial state of the PDFA and the predicate `turnDomain`. Considering the example in Fig. 1a and the PDFA in Fig. 2b, the new initial condition is as follows in PDDL:

```
(:init (and (road 11 21) (road 11 21) ...
  (spare-in 21) (spare-in 12) ...
  (q0 51) (turnDomain)))
```

Listing 2 PDDL initial condition for $\varphi = \Diamond(vAt(51))$

The new goal condition is specified as $G' = \{\bigvee q_i \mid q_i \in F^P\} \cup \{\text{turnDomain}\}$, i.e., we want the PDFA to be in one of its accepting states and `turnDomain`, as follows:

```
(:goal (and (q1 51) (turnDomain)))
```

Listing 3 PDDL goal condition for $\varphi = \Diamond(vAt(51))$

Note that, both in the initial and goal conditions of the new planning problem, PDFA states are grounded back on the objects of interest thanks to the inverse of the mapping m^{obj} .

Executions of a policy for our new FOND planning problem Γ' are $\vec{e}' : [a'_1, t_1, a'_2, t_2, \dots, a'_n, t_n]$, where $a'_i \in A'$ are the real domain actions, and t_1, \dots, t_n are sequences of synchronization `trans` actions, which, at the end, can be easily removed to extract the desired execution $\vec{e} : [a'_1, a'_2, \dots, a'_n]$.

In the remainder of the paper, we refer to the compilation just exposed as FONDforLTLPLTL.

Theoretical property of the PDDL encoding

We now study the theoretical properties of the encoding presented in this section. Theorem 1 states that solving FOND planning for LTL_f/PPLTL goals amounts to solving standard FOND planning problems for reachability goals. A policy for the former can be easily derived from a policy for the latter.

Theorem 1 *Let Γ be a FOND planning problem with an LTL_f/PPLTL goal φ , and Γ' be the compiled FOND planning problem with a reachability goal. Then, Γ has a policy $\pi : (2^{\mathcal{F}})^+ \rightarrow A$ iff Γ' has a policy $\pi' : (2^{\mathcal{F}'})^+ \rightarrow A'$.*

Proof (\rightarrow). We start with a policy π of the original problem that is winning by assumption. Given π , we can always build a new policy, which we call π' , following the encoding presented in Section 3 of the paper. The newly constructed policy will modify histories of π by adding fluents and an auxiliary deterministic action `trans`, both related to the DFA associated with the LTL_f/PPLTL formula φ . Now, we show that π' is an executable policy and that is winning for Γ' . To see the executability, observe that, by construction of the new planning problem Γ' , all action effects Eff_a of the original problem Γ are modified in a way that all action effects of the original problem Γ are not modified and that the auxiliary action `trans` only changes the truth value of additional fluents given by the DFA \mathcal{A}_φ^p (i.e., automaton states). Therefore, the newly constructed policy π' can be executed. To see that π' is winning and satisfies the LTL_f/PPLTL goal formula φ , we reason about all possible executions. For all executions, every time the policy π' stops we can always extract an induced state trajectory of length n such that its last state s'_n will contain one of the final states F^p of the automaton \mathcal{A}_φ^p . This means that the induced state trajectory is accepted by the automaton \mathcal{A}_φ^p . Then, by Theorem [15, 17] $\tau \models \varphi$.

(\leftarrow). From a winning policy π' for the compiled problem, we can always project out all automata auxiliary `trans` actions obtaining a corresponding policy π . We need to show that the resulting policy π is winning, namely, it can be successfully executed on the original problem Γ and satisfies the LTL_f/PPLTL goal formula φ . The executability follows from the fact that the deletion of `trans` actions and related auxiliary fluents from state trajectories induced by π does not modify any precondition/effect of original domain actions (i.e., $a \in \mathcal{A}$). Hence, under the right preconditions, any domain action can be executed. Finally, the satisfaction of the LTL_f/PPLTL formula φ follows directly from Theorem [15, 17]. Indeed, every execution of the winning policy π' stops when reaching one of the final states F^p of the automaton \mathcal{A}_φ^p in the last state s_n , thus every execution of π would satisfy φ . Thus, the thesis holds. \square

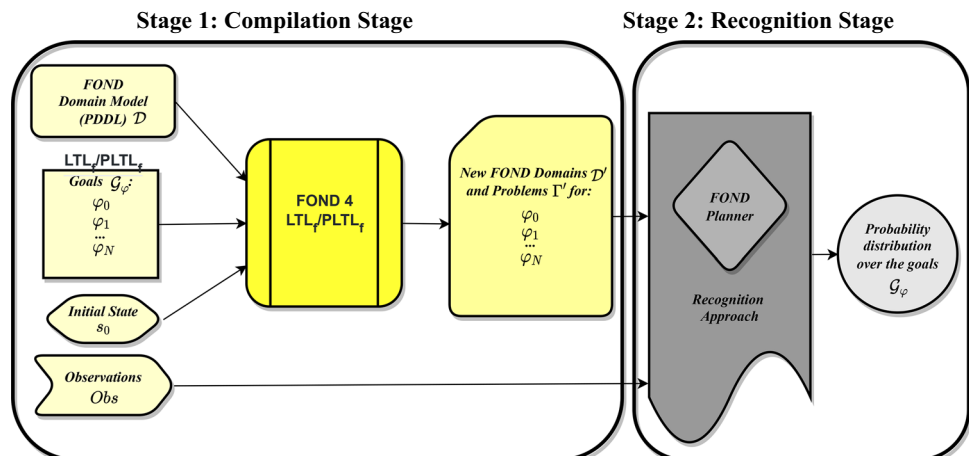
4 Goal recognition in FOND planning domains with LTL_f and PPLTL goals

This section introduces the recognition approach that is able to recognizing temporally extended (LTL_f and PPLTL) goals in FOND planning domains. Our approach extends the probabilistic framework of [35] to compute posterior probabilities over temporally extended goal hypotheses, by reasoning over the set of possible executions of policies π and the observations. This works in two stages: the *compilation stage* and the *recognition stage*. The following sections describe in detail how these two stages work. Figure 3 illustrates how our approach works.

4.1 Goal recognition problem

We define the task of goal recognition in FOND planning domains with LTL_f and PPLTL goals by extending the stan-

Fig. 3 Overview of our solution approach



dard definition of *Plan Recognition as Planning* [34], as follows.

Definition 8 A **temporally extended goal recognition problem** in a FOND planning setting with temporally extended goals (LTL_f and/or PPLTL) is a tuple $\mathcal{T}_\varphi = \langle \mathcal{D}, s_0, \mathcal{G}_\varphi, Obs \rangle$, where: $\mathcal{D} = \langle 2^{\mathcal{F}}, A, \alpha, tr \rangle$ is a FOND planning domain; s_0 is the initial state; $\mathcal{G}_\varphi = \{\varphi_0, \varphi_1, \dots, \varphi_n\}$ is the set of goal hypotheses formalized in LTL_f or PPLTL, including the intended goal $\varphi^* \in \mathcal{G}_\varphi$; $Obs = \langle o_0, o_1, \dots, o_n \rangle$ is a sequence of successfully executed (non-deterministic) actions of a policy π_{φ^*} that achieves the intended goal φ^* , s.t. $o_i \in A$.

Since we deal with non-deterministic domain models, an observation sequence Obs corresponds to a successful execution \vec{e} in the set of all possible executions \vec{E} of a *strong-cyclic policy* π that achieves the actual intended hidden goal φ^* . In this work, we assume two recognition settings: *Offline Keyhole Recognition*, and *Online Recognition*. In *Offline Keyhole Recognition* the observed agent is completely unaware of the recognition process [3], the observation sequence Obs is given at once, and it can be either *full* or *partial*—in a *full observation sequence*, the recognizer has access to all actions of an agent’s plan, whereas, in a *partial observation sequence*, only a sub-sequence thereof. By contrast, in *Online Recognition* [39], the observed agent is also unaware of the recognition process, but the observation sequence is revealed incrementally instead of being given in advance and at once, as in *Offline Recognition*, thus making the recognition process an already much harder task.

An “ideal” solution for a goal recognition problem comprises a selection of the goal hypotheses containing only the single actual intended hidden goal $\varphi^* \in \mathcal{G}$ that the observation sequence Obs of a plan execution achieves [34, 35]. Fundamentally, there is no exact solution for a goal recognition problem, but it is possible to produce a probability distribution over the goal hypotheses and the observations, so that the goals that “best” explain the observation sequence are the most probable ones. A solution to a goal recognition problem in FOND planning with temporally extended goals is defined in Definition 9.

Definition 9 Solving a goal recognition problem \mathcal{T}_φ requires selecting a temporally extended goal hypothesis $\hat{\varphi} \in \mathcal{G}_\varphi$ such that $\hat{\varphi} = \varphi^*$, and it represents how well $\hat{\varphi}$ predicts or explains what observation sequence Obs aims to achieve.

Existing recognition approaches often return either a probability distribution over the set of goals [35, 37], or scores associated with each possible goal hypothesis [31]. Our framework returns a probability distribution \mathbb{P} over the set of temporally extended goals \mathcal{G}_φ that “best” explains the observations sequence Obs .

4.2 Probabilistic goal recognition

The probabilistic framework for *Plan Recognition as Planning* of [35] sets the probability distribution for every goal G in the set of goal hypotheses \mathcal{G} , and the observation sequence Obs to be a Bayesian posterior conditional probability, as follows:

$$\mathbb{P}(G \mid Obs) = \eta * \mathbb{P}(Obs \mid G) * \mathbb{P}(G) \quad (1)$$

where $\mathbb{P}(G)$ is the *a priori* probability assigned to goal G , η is a normalization factor inversely proportional to the probability of Obs , and $\mathbb{P}(Obs \mid G)$ is

$$\mathbb{P}(Obs \mid G) = \sum_{\pi} \mathbb{P}(Obs \mid \pi) * \mathbb{P}(\pi \mid G) \quad (2)$$

$\mathbb{P}(Obs \mid \pi)$ is the probability of obtaining Obs by executing a policy π and $\mathbb{P}(\pi \mid G)$ is the probability of an agent pursuing G to select π . What follows extends the probabilistic framework above to recognize temporally extended goals in FOND planning domain models.

4.3 Compilation stage

We perform a *compilation stage* that allows us to use any off-the-shelf FOND planner to extract policies for temporally extended goals. To this end, we compile and generate new FOND planning domain models Γ' for the set of possible temporally extended goals \mathcal{G}_φ using the compilation approach described in Section 3. Specifically, for every goal $\varphi \in \mathcal{G}_\varphi$, our compilation takes as input a FOND planning problem Γ , where Γ contains the FOND planning domain \mathcal{D} along with an initial state s_0 and a temporally extended goal φ . Finally, as a result, we obtain a new FOND planning problem Γ' associated with the new domain \mathcal{D}' . Note that such a new FOND planning domain Γ' encodes new predicates and transitions that allow us to plan for temporally extended goals by using off-the-shelf FOND planners.

Corollary 1 Let \mathcal{T}_φ be a goal recognition problem over a set of LTL_f/PPLTL goals \mathcal{G}_φ and let \mathcal{T}' be the compiled goal recognition problem over a set of propositional goals \mathcal{G} . Then, if \mathcal{T}' has a set of winning policies that solve the set of propositional goals in \mathcal{G} , then \mathcal{T}_φ has a set of winning policies that solve its LTL_f/PPLTL goals.

Proof It follows from Theorem 1 that a bijective mapping exists between policies of FOND planning for LTL_f/PPLTL goals and policies of standard FOND planning. Therefore, the thesis holds. \square

4.4 Recognition stage

The stage that performs the goal recognition task comprises extracting policies for every goal $\varphi \in \mathcal{G}_\varphi$. From such policies along with observations Obs , we compute posterior probabilities for the goals \mathcal{G}_φ by matching the observations with all possible executions in the set of executions \vec{E} of the policies. To ensure compatibility with the policies, the recognizer assumes knowledge of the preference relation over actions for the observed agent when unrolling the policy during search.

Computing policies and the set of executions \vec{E} for \mathcal{G}_φ

The recognizer extracts policies for every goal $\varphi \in \mathcal{G}_\varphi$ using the new FOND planning domain models Γ' , and for each of these policies, it enumerates the set of possible executions \vec{E} . The aim of enumerating the possible executions \vec{E} for a policy π is to attempt to infer what execution $\vec{e} \in \vec{E}$ the observed agent is performing in the environment. Environmental non-determinism prevents the recognizer from determining the specific execution \vec{e} the observed agent goes through to achieve its goals. The recognizer considers possible executions that are all paths to the goal with no repeated states. The fact that the probability of entering loops multiple times is low partially justifies this assumption, and relaxing it is an important research direction for future work.

After enumerating the set of possible executions \vec{E} for a policy π , we compute the average distance of all actions in the set of executions \vec{E} to a goal φ from initial state s_0 . Note that strong-cyclic solutions may have infinite possible executions. However, here we consider executions that do not enter loops, and for those entering possible loops, we consider only the ones entering loops *at most* once. Indeed, the occurrence of possibly repeated actions does not affect the computation of the average distance. In other words, if the observed agent executes the same action repeatedly often, it does not change its distance to the goal. The average distance aims to estimate “how far” every observation $o \in Obs$ is to goal φ . This average distance is computed because some executions $\vec{e} \in \vec{E}$ may share the same action in execution sequences but at different time steps. We refer to this average distance as \mathbf{d} . For example, consider the policy π depicted in Fig. 1b. This policy π has two possible executions for achieving a goal from the initial state, and these two executions share some actions, such as (move 11 21). In particular, this action appears twice in Fig. 1b due to its uncertain outcome. Therefore, this action has two different distances (if we count the number of remaining actions towards a goal) to the goal: *distance* = 1, if the outcome of this action generates the state s_2 ; and *distance* = 2, if the outcome of this action generates the state s_3 . Hence, since this policy π has two possible executions, and the sum of the distances is 3, the average distance for this action to a goal is $\mathbf{d} = 1.5$. The

average distances for the other actions in this policy are: $\mathbf{d} = 1$ for (changetire 21), because it appears only in one execution; and $\mathbf{d} = 0$ for (move 21 22), because the execution of this action achieves a goal.

We use \mathbf{d} to compute an *estimated score* that expresses “how far” every observed action in the observation sequence Obs is to a temporally extended goal φ in comparison to the other goals in the set of goal hypotheses \mathcal{G}_φ . This means that the goal(s) with the lowest score(s) along the execution of the observed actions $o \in Obs$ is (are) the one(s) that, most likely, the observation sequence Obs aims to achieve. Note that, the average distance \mathbf{d} for those observations $o \in Obs$ that are not in the set of executions \vec{E} of a policy π , is set to a large constant number, i.e., to $\mathbf{d} = e^5$. As part of the computation of this *estimated score*, we compute a *penalty value* that directly affects the *estimated score*. This *penalty value* represents a penalization that aims to increase the *estimated score* for those goals in which each pair of subsequent observations $\langle o_{i-1}, o_i \rangle$ in Obs does not have any relation of order in the set of executions \vec{E} of these goals. We use the Euler constant e to compute this *penalty value*, formally defined as $e^{\mathbf{p}(o_{i-1}, o_i)}$, in which $\mathcal{R}(\vec{e})$ is the set of order relation of an execution \vec{e} , where

$$\mathbf{p}(o_{i-1}, o_i) = \begin{cases} 1, & \text{if } \{\forall \vec{e} \in E \mid \langle o_{i-1} < o_i \rangle \notin \mathcal{R}(\vec{e})\} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Equation (4) formally defines the computation of the *estimated score* for every goal $\varphi \in \mathcal{G}_\varphi$ given a pair of subsequent observations $\langle o_{i-1}, o_i \rangle$, and the set of goal hypotheses \mathcal{G}_φ .

$$\frac{e^{\mathbf{p}(o_{i-1}, o_i)} * \mathbf{d}(o_i, \varphi)}{\sum_{\varphi' \in \mathcal{G}_\varphi} \mathbf{d}(o_i, \varphi')} \quad (4)$$

Example 2 To exemplify the computation of the *estimated score* for every goal $\varphi \in \mathcal{G}_\varphi$, consider the recognition problem in Fig. 4: s_0 is $vAt(11)$; the goal hypotheses \mathcal{G}_φ are expressed as LTL_f goals, such that $\varphi_0 = \Diamond vAt(51)$, $\varphi_1 = \Diamond vAt(33)$, and $\varphi_2 = \Diamond vAt(15)$; $Obs = \{o_0 : (\text{move } 11 \ 21), o_1 : (\text{changetire } 22)\}$. The intended goal φ^* is φ_1 . Before computing the *estimated score* for the goals, we first perform the compilation process presented before. Afterward, we extract policies for every goal $\varphi \in \mathcal{G}_\varphi$, enumerate the possible executions \vec{E} for the goals \mathcal{G}_φ from the extracted policies, and then compute the average distance \mathbf{d} of all actions in the set of executions \vec{E} for the goals \mathcal{G}_φ from s_0 . The number of possible executions \vec{E} for the goals are: $\varphi_0 : |\vec{E}| = 8$, $\varphi_1 : |\vec{E}| = 8$, and $\varphi_2 : |\vec{E}| = 16$. The average distances \mathbf{d} of all actions in \vec{E} for the goals are as follows:

- φ_0 : (move 11 21) = 4.5, (changetire 21) = 4, (move 21 31) = 3, (changetire 31) = 2.5,

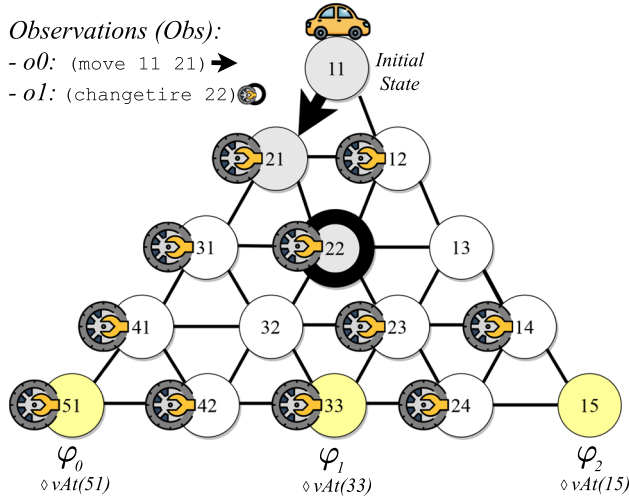


Fig. 4 Recognition problem example

- (move 31 41) = 1.5, (changetire 41) = 1, (move 41 51) = 0;
- φ_1 : (move 11 21) = 4.5, (changetire 21) = 4, (move 21 22) = 3, (changetire 22) = 2.5, (move 22 23) = 1.5, (changetire 23) = 1, (move 23 33): 0;
- φ_2 : (move 11 21) = 6, (changetire 21) = 5.5, (move 21 22) = 4.5, (changetire 22) = 4, (move 22 23) = 3, (changetire 23) = 2.5, (changetire 24) = 1, (move 23 24) = 1.5, (move 24 15) = 0.

Once having the average distances \mathbf{d} of the actions in \vec{E} for all goals, we can then compute the *estimated score* for \mathcal{G}_φ for every observation $o \in Obs$: o_0 (move 11 21) : $\varphi_0 = \frac{4.5}{4.5+6} = 0.43$, $\varphi_1 = \frac{4.5}{4.5+6} = 0.43$, $\varphi_2 = \frac{6}{4.5+6} = 0.57$; and o_1 (changetire 22) : $\varphi_0 = \frac{e^1 * e^5}{6.5} = 61.87$, $\varphi_1 = \frac{2.5}{e^5+2.5} = 0.016$, $\varphi_2 = \frac{4}{e^5+4} = 0.026$. Note that for the observation o_1 , the average distance \mathbf{d} for φ_0 is $e^5 = 148.4$ because this observation is not an action for one of the executions in the set of executions for this goal (Obs aims to achieve the intended goal $\varphi^* = \varphi_1$). Furthermore, the *penalty value* is applied to φ_0 , i.e., $e^1 = 2.71$. It is possible to see that the *estimated score* of the intended goal φ_1 is always the lowest for all observations Obs , especially when observing the second observation o_1 . Note that our approach correctly infers the intended goal φ^* , even when observing with just few actions.

Computing posterior probabilities for \mathcal{G}_φ

To compute the posterior probabilities over the set of possible temporally extended goals \mathcal{G}_φ , we start by computing the *average estimated score* for every goal $\varphi \in \mathcal{G}_\varphi$ for every

observation $o \in Obs$, and we formally define this computation as $\mathcal{E}(\varphi, Obs, \mathcal{G}_\varphi)$, as follows:

$$\mathcal{E}(\varphi, Obs, \mathcal{G}_\varphi) = \left(\frac{\sum_{i=0}^{|Obs|} \frac{e^{\mathbf{p}(o_{i-1}, o_i)} * \mathbf{d}(o_i, \varphi)}{\sum_{\varphi' \in \mathcal{G}_\varphi} \mathbf{d}(o_i, \varphi')}}{|Obs|} \right) \quad (5)$$

The *average estimated score* \mathcal{E} aims to estimate “how far” a goal φ is to be achieved compared to other goals ($\mathcal{G}_\varphi \setminus \{\varphi\}$) averaging among all the observations in Obs . The lower the *average estimated score* \mathcal{E} to a goal φ , the more likely such a goal is to be the one that the observed agent aims to achieve. Consequently, \mathcal{E} has two important properties defined in (5), as follows.

Proposition 1 *Given that the sequence of observations Obs corresponds to an execution $\vec{e} \in \vec{E}$ that aims to achieve the actual intended hidden goal $\varphi^* \in \mathcal{G}_\varphi$, the average estimated score outputted by \mathcal{E} will tend to be the lowest for φ^* in comparison to the scores of the other goals ($\mathcal{G}_\varphi \setminus \{\varphi^*\}$), as observations increase in length.*

Proposition 2 *If we restrict the recognition setting and define that the goal hypotheses \mathcal{G}_φ are not sub-goals of each other, and observe all observations in Obs (i.e., full observability), we will have the intended goal φ^* with the lowest score among all goals, i.e., $\forall \varphi \in \mathcal{G}_\varphi$ is the case that $\mathcal{E}(\varphi^*, Obs, \mathcal{G}_\varphi) \leq \mathcal{E}(\varphi, Obs, \mathcal{G}_\varphi)$.*

After defining the computation of the *average estimated score* \mathcal{E} for the goals using (5), we can define how our approach tries to maximize the probability of observing a sequence of observations Obs for a given goal φ , as follows:

$$\mathbb{P}(Obs | \varphi) = [1 + \mathcal{E}(\varphi, Obs, \mathcal{G}_\varphi)]^{-1} \quad (6)$$

Thus, by using the *estimated score* in (6), we can infer that the goals $\varphi \in \mathcal{G}_\varphi$ with the lowest *estimated score* will be the most likely to be achieved according to the probability interpretation from (5). For instance, consider the goal recognition problem presented in Example 2, and the *estimated scores* we computed for the temporally extended goals φ_0 , φ_1 , and φ_2 based on the observation sequence Obs . From this, we have the following probabilities $\mathbb{P}(Obs | \varphi)$ for the goals:

- $\mathbb{P}(Obs | \varphi_0) = [1 + (31.15)]^{-1} = 0.03$
- $\mathbb{P}(Obs | \varphi_1) = [1 + (0.216)]^{-1} = 0.82$
- $\mathbb{P}(Obs | \varphi_2) = [1 + (0.343)]^{-1} = 0.74$

After normalizing the probabilities using the normalization factor η^2 , and assuming that the prior probability $\mathbb{P}(\varphi)$

$$\eta^2 = \frac{1}{[\sum_{\varphi \in \mathcal{G}_\varphi} \mathbb{P}(Obs | \varphi) * \mathbb{P}(\varphi)]^{-1}}$$

is equal to every goal in the set of goals \mathcal{G}_φ , (6) computes the posterior probabilities (1) for the temporally extended goals \mathcal{G}_φ . A *solution* to a recognition problem \mathcal{T}_φ (Definition 8) is a set of temporally extended goals \mathcal{G}_φ^* with the *maximum probability*: $\mathcal{G}_\varphi^* = \arg \max_{\varphi \in \mathcal{G}_\varphi} \mathbb{P}(\varphi \mid Obs)$. Hence, considering the normalizing factor η and the probabilities $\mathbb{P}(Obs \mid \varphi)$ computed before, we then have the following posterior probabilities for the goals in Example 2: $\mathbb{P}(\varphi_0 \mid Obs) = 0.001$; $\mathbb{P}(\varphi_1 \mid Obs) = 0.524$; and $\mathbb{P}(\varphi_2 \mid Obs) = 0.475$. Recall that in Example 2, φ^* is φ_1 , and according to the computed posterior probabilities, we then have $\mathcal{G}_\varphi^* = \{\varphi_1\}$, so our approach yields only the intended goal by observing just two observations.

Using the *average distance* \mathbf{d} and the *penalty value* \mathbf{p} allows our approach to disambiguate similar goals during the recognition stage. For instance, consider the following possible temporally extended goals: $\varphi_0 = \phi_1 \mathcal{U} \phi_2$ and $\varphi_1 = \phi_2 \mathcal{U} \phi_1$. Here, both goals have the same formulas to be achieved, i.e., ϕ_1 and ϕ_2 , but in a different order. Thus, even having the same formulas to be achieved, the sequences of their policies' executions are different. Therefore, the average distances are also different, possibly a smaller value for the temporally extended goal that the agent aims to achieve, and the penalty value may also be applied to the other goal if two subsequent observations do not have any order relation in the set of executions for this goal.

Computational analysis

The most expensive computational part of our recognition approach is computing the policies π for the goal hypotheses \mathcal{G}_φ . Thus, our approach requires $|\mathcal{G}_\varphi|$ calls to an off-the-shelf FOND planner. Hence, the computational complexity of our recognition approach is linear in the number of goal hypotheses $|\mathcal{G}_\varphi|$. In contrast, to recognize goals and plans in *Classical Planning* settings, the approach of [35] requires $2 * |\mathcal{G}|$ calls to an off-the-shelf *Classical* planner. Concretely, to compute $\mathbb{P}(Obs \mid G)$, Ramirez and Geffner's approach computes two plans for every goal and based on these two plans, they compute a *cost-difference* between these plans and plug it into a Boltzmann equation. For computing these two plans, this approach requires a non-trivial transformation process that modifies both the domain and problem, i.e., an augmented domain and problem that compute a plan that *complies* with the observations, and another augmented domain and problem to compute a plan that *does not comply* with the observations. Essentially, the intuition of Ramirez and Geffner's approach is that the lower the *cost-difference* for a goal, the higher the probability for this goal, much similar to the intuition of our *estimated score* \mathcal{E} .

5 Experiments and evaluation

This section details experiments and evaluations carried out to validate the effectiveness of our recognition approach. The empirical evaluation covers thousands of goal recognition problems using well-known FOND planning domain models with different types of temporally extended goals expressed in LTL_f and PPLTL.

The source code of our PDDL encoding for LTL_f and PPLTL goals³ and our temporally extended goal recognition approach⁴, as well as the recognition datasets and results are available on GitHub.

5.1 Domains, recognition datasets, and setup

The experiments and evaluation analysis employ six different well-known FOND planning domain models: BLOCKS-WORLD, LOGISTICS, TIDY-UP, TIREWORLD, TRIANGLE-TIREWORLD, and ZENO-TRAVEL. Most of them are commonly used in the AI Planning community to evaluate FOND planners [23, 25]. The domain models involve practical real-world applications, such as navigating, stacking, picking up and putting down objects, loading and unloading objects, loading and unloading objects, and etc. Some domains combine more than one of the characteristics above, namely, LOGISTICS, TIDY-UP [26], and ZENO-TRAVEL, which involve navigating and manipulating objects in the environment. In practice, our recognition approach is capable of recognizing not only the set of facts of a goal that an observed agent aims to achieve from a sequence of observations, but also the *temporal order* (e.g., *exact order*) in which the agent aims to achieve this set of facts that represents a temporally extended goal. For instance, for TIDY-UP, is a real-world application domain, in which the purpose is defining planning tasks for a household robot that could assist elder people in smart-home application, our approach would be able to monitor and assist the household robot to achieve its goals in a specific order.

Based on these FOND planning domain models, we build different recognition datasets: a *baseline* dataset using conjunctive goals ($\phi_1 \wedge \phi_2$) and datasets with LTL_f and PPLTL goals.

The LTL_f datasets use three types of goals:

- $\Diamond \phi$, where ϕ is a propositional formula expressing that *eventually* ϕ will be achieved. This temporal formula is analogous to a reachability goal;

³ <https://github.com/whitemech/FOND4LTLf>

⁴ https://github.com/ramonpereira/goal-recognition-ltlf_pltlf-fond

- $\Diamond(\phi_1 \wedge \bigcirc(\Diamond\phi_2))$, expressing that ϕ_1 must hold before ϕ_2 holds. For instance, we can define a temporal goal that expresses the order in which a set of packages in LOGISTICS domain should be delivered;
- $\phi_1 \mathcal{U} \phi_2$: ϕ_1 must hold *until* ϕ_2 is achieved. For the TIDY-UP domain, we can define a temporal goal that no one can be in the kitchen until the robot cleans the kitchen.

The PPLTL datasets use two types of goals:

- $\phi_1 \wedge \Diamond\phi_2$, expressing that ϕ_1 holds and ϕ_2 held once. For instance, in the BLOCKS- WORLD domain, we can define a past temporal goal that only allows stacking a set of blocks (a, b, c) once another set of blocks has been stacked (d, e);
- $\phi_1 \wedge (\neg\phi_2 \mathcal{S} \phi_3)$, expressing that the formula ϕ_1 holds and *since* ϕ_3 held ϕ_2 was not true anymore. For instance, in ZENO- TRAVEL, we can define a past temporal goal expressing that person1 is at city1 and since the person2 is at city1, the aircraft must not pass through city2 anymore.

Thus, in total, there are six different recognition datasets over the six FOND planning domains and temporal formulas presented above. Each of these datasets contains hundreds of recognition problems (≈ 390 recognition problems per dataset), such that each recognition problem \mathcal{T}_φ in these datasets is comprised of a FOND planning domain model \mathcal{D} , an initial state s_0 , a set of possible goals \mathcal{G}_φ (expressed in either LTL_f or PPLTL), the actual intended hidden goal in the set of possible goals $\varphi^* \in \mathcal{G}_\varphi$, and the observation sequence Obs . Note that the set of possible goals \mathcal{G}_φ contains very similar goals (i.e., $\varphi_0 = \phi_1 \mathcal{U} \phi_2$ and $\varphi_1 = \phi_2 \mathcal{U} \phi_1$), and all possible goals can be achieved from the initial state by a strong-cyclic policy. For instance, for the TIDY- UP domain, we define the following LTL_f goals as possible goals \mathcal{G}_φ :

- $\varphi_0 = \Diamond((\text{wiped desk1}) \wedge \bigcirc(\Diamond(\text{on book1 desk1})))$;
- $\varphi_1 = \Diamond((\text{on book1 desk1}) \wedge \bigcirc(\Diamond(\text{wiped desk1})))$;
- $\varphi_2 = \Diamond((\text{on cup1 desk2}) \wedge \bigcirc(\Diamond(\text{wiped desk2})))$;
- $\varphi_3 = \Diamond((\text{wiped desk2}) \wedge \bigcirc(\Diamond(\text{on cup1 desk2})))$;

Note that some of the goals described above share the same formulas and fluents, but some of these formulas must be achieved in a different order, e.g., φ_0 and φ_1 , and φ_2 and φ_3 . Note that our recognition approach is very accurate in discerning (Table 1) the order that the intended goal aims to be achieved based on few observations (executions of the agent in the environment).

As mentioned earlier in the paper, an observation sequence contains a sequence of actions that represent an execution \vec{e}

in the set of possible executions \vec{E} of policy π that achieves the actual intended hidden goal φ^* , and as before, this observation sequence Obs can be full or partial. To generate the observations Obs for φ^* and build the recognition problems, our approach extracts strong-cyclic policies using different FOND planners, such as PRP and MyND. A full observation sequence represents an execution (a sequence of executed actions) of a strong-cyclic policy that achieves the actual intended hidden goal φ^* , i.e., 100% of the actions of \vec{e} being observed. A partial observation sequence is represented by a sub-sequence of actions of a full execution that aims to achieve the actual intended hidden goal φ^* (e.g., an execution with “missing” actions, due to a sensor malfunction). In our recognition datasets, we define four levels of observability for a partial observation sequence: 10%, 30%, 50%, or 70% of its actions being observed. For instance, for a full observation sequence Obs with 10 actions (100% of observability), a corresponding partial observations sequence with 10% of observability would have only one observed action, and for 30% of observability three observed actions, and so on for the other levels of observability.

We ran all experiments using PRP [25] planner with a single core of a 12 core Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz with 16GB of RAM, set a maximum memory usage limit of 8GB, and set a 10-minute timeout for each recognition problem. We are unable to provide a *direct comparison* of our approach against existing recognition approaches in the literature because most of these approaches perform a non-trivial process that transforms a recognition problem into planning problems to be solved by a planner [35, 37]. Even adapting such a transformation to work in FOND settings with temporally extended goals, one cannot guarantee that it will work properly in the problem setting introduced in this paper.

5.2 Evaluation metrics

Our evaluation uses widely known metrics in the *Goal and Plan Recognition* literature [31, 34, 39]. To evaluate our approach in the *Offline Keyhole Recognition* setting, we use four metrics, as follows:

- *True Positive Rate (TPR)* measures the fraction of times that the intended hidden goal φ^* was correctly recognized, e.g., the percentage of recognition problems that our approach correctly recognized the intended goal. A **higher** TPR indicates better accuracy, measuring how often the intended hidden goal had the highest probability $P(\varphi \mid Obs)$ among the possible goals. TPR (7) is the

Table 1 Offline Recognition results for Conjunctive, LTL_f, and PPLTL goals

	$ \mathcal{G}_\phi $	$ \text{Obs} $	Time	TPR	FPR	FNR	F1-Score	Time	TPR	FPR	FNR	F1-Score
Conjunctive Goals				LTL _f Eventuality Goals								
$\phi_1 \wedge \phi_2$				$\Diamond\phi$								
10	5.2	3.85	189.1	0.75	0.15	0.25	0.63	243.8	0.74	0.11	0.26	0.60
30		10.7	187.2	0.85	0.08	0.15	0.78	235.1	0.86	0.10	0.14	0.78
50		17.4	188.4	0.83	0.09	0.17	0.82	242.1	0.89	0.07	0.11	0.92
70		24.3	187.8	0.86	0.08	0.14	0.84	232.1	0.92	0.08	0.08	0.87
100		34.7	190.4	0.85	0.09	0.15	0.86	272.8	0.95	0.09	0.05	0.90
LTL _f Ordering Goals				LTL _f Goals Until								
$\Diamond(\phi_1 \wedge \bigcirc(\Diamond\phi_2))$				$\phi_1 \mathcal{U} \phi_2$								
10	4.0	2.1	136.1	0.68	0.15	0.32	0.62	217.9	0.79	0.11	0.21	0.72
30		5.4	130.9	0.84	0.13	0.16	0.76	215.8	0.91	0.12	0.09	0.82
50		8.8	132.1	0.88	0.10	0.12	0.80	210.1	0.93	0.10	0.07	0.83
70		12.5	129.2	0.95	0.06	0.05	0.89	211.5	0.97	0.09	0.03	0.86
100		17.1	126.6	0.94	0.05	0.06	0.90	207.7	0.97	0.07	0.03	0.87
PPLTL Goals Once				PPLTL Goals Since								
$\phi_1 \wedge \Diamond\phi_2$				$\phi_1 \wedge (\neg\phi_2 \mathcal{S} \phi_3)$								
10	4.0	1.7	144.8	0.73	0.11	0.27	0.67	173.5	0.76	0.18	0.24	0.64
30		4.6	141.3	0.84	0.07	0.16	0.79	173.3	0.87	0.12	0.13	0.78
50		7.3	141.9	0.89	0.08	0.11	0.82	172.9	0.85	0.09	0.15	0.79
70		10.3	142.9	0.95	0.07	0.05	0.87	171.1	0.97	0.07	0.03	0.91
100		14.2	155.8	0.97	0.07	0.03	0.88	169.3	0.94	0.02	0.06	0.93

ratio between true positive results⁵, and the sum of true positive and false negative results⁶;

$$TPR = \frac{TP}{TP + FN} = 1 - FNR \quad (7)$$

- **False Positive Rate (FPR)** is a metric that measures how often goals other than the intended goal are recognized (wrongly) as the intended ones. A **lower FPR** indicates better accuracy. FPR is the ratio between false positive results⁷, and the sum of false positive and true negative results⁸;

$$FPR = \frac{FP}{FP + TN} \quad (8)$$

- **False Negative Rate (FNR)** aims to measure the fraction of times in which the intended correct goal was recog-

nized incorrectly. A **lower FNR** indicates better accuracy. FNR (9) is the ratio between false negative results and the sum of false negative and true positive results;

$$FNR = \frac{FN}{FN + TP} = 1 - TPR \quad (9)$$

- **F1-Score (10)** is the harmonic mean of precision and sensitivity (i.e., TPR), representing the trade-off between true positive and false positive results. The **highest possible value** of an F1-Score is 1.0, indicating perfect precision and sensitivity, and the **lowest possible value** is 0. Thus, **higher F1-Score** values indicate better accuracy.

$$F1 - Score = \frac{2 * TP}{2TP + FP + FN} \quad (10)$$

In contrast, to evaluate our approach in the *Online Recognition* setting, we use the following metric:

- **Ranked First** is a metric that measures the number of times the intended goal hypothesis φ^* has been correctly ranked first as the most likely intended goal, and **higher** values for this metric indicate better accuracy for performing online recognition.

⁵ True positive results represent the number of goals that has been recognized correctly.

⁶ False negative results represent the number of correct goals that has not been recognized.

⁷ False positive results are the number of incorrect goals that has been recognized as the correct ones.

⁸ True negative results represent the number of incorrect goals has been recognized correctly as the incorrect ones.

In addition to the metrics mentioned above, we also evaluate our recognition approach in terms of *recognition time* (*Time*), which is the average time in seconds to perform the recognition process (including the calls to a FOND planner);

5.3 Offline keyhole recognition results

We now assess how accurate our recognition approach is in the *Keyhole Recognition* setting. Table 1 shows three inner tables that summarize and aggregate the average results of all the six datasets for four different metrics, such as *Time*, *TPR*, *FPR*, and *FNR*. $|\mathcal{G}_\phi|$ represents the average number of goals in the datasets, and $|Obs|$ the average number of observations. Each row in these inner tables represents the observation level, varying from 10% to 100%. Figure 5 shows the performance of our approach by comparing the results using *F1-Score* for the six types of temporal formulas we used for evaluation. Table 2 shows in much more detail the results for each of the six datasets that have been used for evaluating of our recognition approach.

Offline results for conjunctive and eventuality goals

The first inner table shows the average results comparing the performance of our approach between conjunctive goals and temporally extended goals using the *eventually* temporal operator \Diamond . We refer to this comparison as the *baseline* since these two types of goals have the same semantics. We can see that the results for these two types of goals are very similar for all metrics. Moreover, it is also possible to see that our recognition approach is very accurate and performs well at all levels of observability, yielding high *TPR* values and low *FPR* and *FNR* values for more than 10% of observability. Note that for 10% of observability, and LTL_f goals for $\Diamond\phi$, the *TPR* average value is 0.74, and it means for 74% of the recognition problems our approach recognized correctly the intended temporally extended goal when observing, on average, only 3.85 actions. Figure 5a shows that our approach yields higher *F1-Score* values (i.e., greater than 0.79) for

these types of formulas when dealing with more than 50% of observability.

Offline results for LTL_f goals

Regarding the results for the two types of LTL_f goals (second inner table), it is possible to see that our approach shows to be accurate for all metrics at all levels of observability, apart from the results for 10% of observability for LTL_f goals in which the formulas must be recognized in a certain order. Note that our approach is accurate even when observing just a few actions (2.1 for 10% and 5.4 for 30%), but not as accurate as for more than 30% of observability. Figure 5b shows that our approach yields higher *F1-Score* values (i.e., greater than 0.75) when dealing with more than 30% of observability.

Offline results for PPLTL goals

Finally, as for the results for the two types of PPLTL goals, it is possible to observe in the last inner table that the overall average number of observations $|Obs|$ is less than the average for the other datasets, making the task of goal recognition more difficult for the PPLTL datasets. Yet, our recognition approach remains accurate when dealing with fewer observations. Moreover, the values of *FNR* increase for low observability, but the *FPR* values are, on average, inferior to ≈ 0.15 . Figure 5c shows that our approach gradually increases the *F1-Score* values when also increases the percentage of observability.

5.4 Online recognition results

With the experiments and evaluation in the *Keyhole Offline* recognition setting in place, we now proceed to present the experiments and evaluation in the *Online* recognition setting. As before, performing the recognition task in the *Online* recognition setting is usually harder than in the offline setting, as the recognition task has to be performed incrementally and gradually, and the recognizer sees the observations step-by-

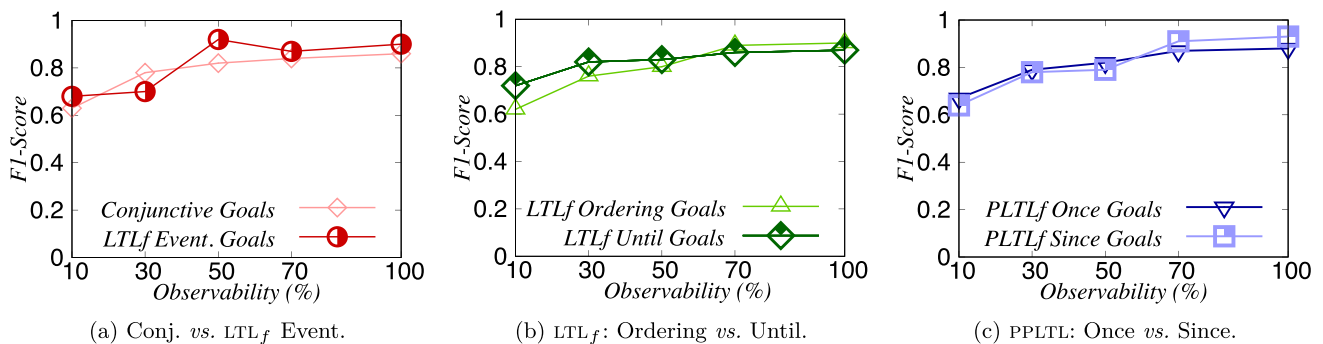


Fig. 5 F1-Score comparison

step, rather than performing the recognition task by analyzing all observations at once, as in the offline recognition setting.

Figure 6 exemplifies the evaluation in the *Online* recognition setting. This uses the *Ranked First* metric, which measures how many times over the observation sequence the correct intended goal φ^* has been ranked first as the *top-1* goal over the goal hypotheses \mathcal{G}_φ . The recognition problem example depicted in Fig. 6 has five goal hypotheses (y-axis), and ten actions in the observation sequence (x-axis). As stated

before, the recognition task in the *Online* setting is done gradually, step-by-step, so at every step our approach essentially ranks the goals according to the probability distribution over the goal hypotheses \mathcal{G}_φ . The example in Fig. 6 shows the correct goal φ^* *Ranked First* six times (at the observation indexes: 4, 6, 7, 8, 9, and 10) over the observation sequence with ten observation, so it means that the goal correct intended goal \mathcal{G}_φ is *Ranked First* (i.e., as the *top-1*, with the highest probability among the goal hypotheses \mathcal{G}_φ) 60%

Table 2 Offline Recognition experimental results for all six FOND domains separately

Conjunctive Goals							LTL _f Eventuality Goals						LTL _f Ordering					
$\phi_1 \wedge \phi_2$							$\Diamond\phi$						$\Diamond(\phi_1 \wedge \bigcirc(\Diamond\phi_2))$					
$ \mathcal{G}_\varphi $	$ Obs $	Time	TPR	FPR	FNR		$ \mathcal{G}_\varphi $	$ Obs $	Time	TPR	FPR	FNR	$ \mathcal{G}_\varphi $	$ Obs $	Time	TPR	FPR	FNR
BLOCKS- WORLD																		
10	6.0	3.92	33.53	0.81	0.10	0.19	6.0	3.92	81.16	0.81	0.06	0.19	4.0	1.33	38.05	0.67	0.10	0.33
30		10.33	33.91	0.86	0.09	0.14		10.58	69.65	0.81	0.10	0.19		3.17	39.49	0.83	0.07	0.17
50		16.67	33.90	0.75	0.14	0.25		17.08	67.16	0.72	0.09	0.28		4.67	39.63	0.89	0.07	0.11
70		23.58	33.98	0.75	0.13	0.25		23.92	67.71	0.78	0.09	0.22		6.67	38.92	0.89	0.07	0.11
100		33.00	34.01	0.75	0.17	0.25		33.58	68.82	0.75	0.14	0.25		9.00	38.28	0.83	0.08	0.17
LOGISTICS																		
10	4.0	3.00	260.92	0.85	0.11	0.15	4.0	3.00	412.34	0.70	0.11	0.30	4.0	2.33	498.70	0.72	0.22	0.28
30		8.11	258.60	0.89	0.08	0.11		8.11	360.49	0.96	0.10	0.04		5.83	468.36	0.89	0.17	0.11
50		13.11	258.58	0.89	0.09	0.11		13.00	383.45	1.00	0.09	0.00		9.17	480.20	0.89	0.19	0.11
70		18.33	251.51	0.96	0.09	0.04		18.11	380.51	1.00	0.09	0.00		13.00	466.86	1.00	0.11	0.00
100		25.44	251.27	1.00	0.08	0.00		25.22	444.65	1.00	0.08	0.00		17.83	450.51	1.00	0.08	0.00
TIDYUP																		
10	4.0	6.56	180.17	0.37	0.27	0.63	4.0	7.00	230.36	0.52	0.20	0.48	4.0	3.50	106.87	0.67	0.28	0.33
30		18.78	178.05	0.48	0.19	0.52		20.00	228.34	0.63	0.32	0.37		9.50	105.93	0.67	0.36	0.33
50		31.00	179.45	0.44	0.22	0.56		32.89	191.61	0.81	0.22	0.19		15.50	105.84	0.78	0.28	0.22
70		43.56	178.79	0.41	0.22	0.59		46.11	193.81	0.81	0.31	0.19		21.83	104.01	0.89	0.19	0.11
100		61.56	179.52	0.33	0.25	0.67		65.33	247.09	1.00	0.28	0.00		30.50	100.16	0.83	0.12	0.17
TIREWORLD																		
10	5.5	1.50	16.88	1.00	0.29	0.00	5.5	1.50	29.17	1.00	0.19	0.00	3.5	1.50	12.59	0.67	0.16	0.33
30		3.50	17.19	1.00	0.04	0.00		3.50	26.39	1.00	0.07	0.00		3.50	11.90	0.72	0.16	0.28
50		6.00	17.35	1.00	0.01	0.00		6.00	21.97	1.00	0.04	0.00		5.67	10.18	0.89	0.05	0.11
70		8.50	17.31	1.00	0.01	0.00		8.50	20.30	1.00	0.00	0.00		7.83	10.20	0.94	0.02	0.06
100		11.50	17.28	1.00	0.00	0.00		11.50	20.97	1.00	0.00	0.00		10.50	10.15	1.00	0.00	0.00
TRIANGLE- TIREWORLD																		
10	3.75	1.67	16.56	0.64	0.16	0.36	3.75	2.08	34.57	0.69	0.13	0.31	4.0	1.67	14.42	0.44	0.14	0.56
30		4.67	16.90	0.86	0.03	0.14		5.58	31.76	0.86	0.03	0.14		3.83	14.57	0.94	0.01	0.06
50		7.33	17.08	0.89	0.03	0.11		8.83	30.83	0.92	0.02	0.08		6.17	14.76	0.83	0.04	0.17
70		10.00	17.12	1.00	0.00	0.00		12.08	32.81	1.00	0.00	0.00		8.50	17.43	1.00	0.00	0.00
100		13.67	17.16	1.00	0.00	0.00		16.67	30.93	1.00	0.00	0.00		11.33	24.17	1.00	0.00	0.00
ZENO- TRAVEL																		
10	7.5	5.67	556.36	0.89	0.08	0.11	7.5	5.33	607.20	0.81	0.05	0.19	4.0	2.67	145.81	0.94	0.01	0.06
30		16.25	549.34	1.00	0.04	0.00		15.17	619.13	0.94	0.02	0.06		7.50	145.33	1.00	0.00	0.00
50		26.50	554.09	1.00	0.02	0.00		24.75	670.07	0.97	0.02	0.03		11.67	141.81	1.00	0.00	0.00
70		37.50	556.76	1.00	0.03	0.00		34.92	619.24	1.00	0.02	0.00		16.67	138.04	1.00	0.00	0.00
100		53.00	569.43	1.00	0.02	0.00		49.42	735.20	1.00	0.02	0.00		23.17	136.37	1.00	0.00	0.00

Table 2 continued

LTL _f Goals Until $\phi_1 \cup \phi_2$							PPLTL Goals Once $\phi_1 \wedge \Diamond \phi_2$						PPLTL Goals Since $\phi_1 \wedge (\neg \phi_2 \mathcal{S} \phi_3)$					
$ \mathcal{G}_\varphi $	$ Obs $	Time	TPR	FPR	FNR		$ \mathcal{G}_\varphi $	$ Obs $	Time	TPR	FPR	FNR	$ \mathcal{G}_\varphi $	$ Obs $	Time	TPR	FPR	FNR
BLOCKS- WORLD																		
10	4.0	1.00	342.01	0.72	0.08	0.28	4.0	1.17	38.97	0.89	0.08	0.11	4.0	1.00	50.98	0.89	0.14	0.11
30		2.83	357.48	0.94	0.01	0.06		3.17	39.15	1.00	0.03	0.00		2.83	53.42	1.00	0.00	0.00
50		3.83	349.43	1.00	0.00	0.00		4.67	38.32	1.00	0.06	0.00		3.83	50.67	1.00	0.00	0.00
70		5.83	355.37	1.00	0.00	0.00		6.67	38.16	1.00	0.07	0.00		5.83	48.33	1.00	0.00	0.00
100		7.67	393.83	1.00	0.00	0.00		9.17	37.92	1.00	0.08	0.00		7.67	47.21	1.00	0.00	0.00
LOGISTICS																		
10	4.0	1.83	310.16	1.00	0.12	0.00	4.0	1.67	554.45	0.78	0.14	0.22	4.0	1.00	643.01	0.83	0.14	0.17
30		4.67	292.87	0.94	0.32	0.06		4.17	541.31	0.94	0.12	0.06		2.33	645.55	0.83	0.18	0.17
50		7.67	282.77	1.00	0.31	0.00		6.33	542.76	0.94	0.12	0.06		3.17	652.51	0.89	0.15	0.11
70		11.00	285.97	1.00	0.28	0.00		9.00	552.88	1.00	0.08	0.00		4.50	648.61	1.00	0.05	0.00
100		14.83	232.00	1.00	0.17	0.00		12.50	630.17	1.00	0.08	0.00		6.00	644.07	1.00	0.00	0.00
TIDYUP																		
10	4.0	3.17	45.20	0.72	0.22	0.28	4.0	3.33	108.24	0.67	0.11	0.33	4.0	3.50	47.25	0.50	0.31	0.50
30		8.33	45.53	0.78	0.26	0.22		9.00	104.46	0.61	0.14	0.39		10.33	46.47	0.56	0.21	0.44
50		13.50	43.40	0.89	0.25	0.11		14.50	105.02	0.72	0.17	0.28		17.00	45.71	0.33	0.24	0.67
70		19.33	43.17	0.94	0.24	0.06		20.33	106.25	0.89	0.19	0.11		23.67	48.27	0.83	0.08	0.17
100		26.83	43.97	1.00	0.17	0.00		28.50	107.56	1.00	0.21	0.00		33.50	48.22	0.67	0.08	0.33
TIREWORLD																		
10	3.5	1.17	5.11	0.72	0.08	0.28	4.0	1.33	6.85	0.56	0.19	0.44	4.0	1.17	18.00	0.67	0.12	0.33
30		3.17	5.17	0.94	0.03	0.06		3.50	6.95	0.83	0.10	0.17		3.17	18.69	0.89	0.07	0.11
50		4.83	5.10	0.94	0.01	0.06		5.50	6.87	0.83	0.10	0.17		4.83	19.09	0.94	0.04	0.06
70		6.50	5.16	1.00	0.06	0.00		7.67	6.82	0.83	0.07	0.17		6.50	19.23	1.00	0.04	0.00
100		9.00	5.55	1.00	0.08	0.00		9.00	5.55	1.00	0.08	0.00		9.00	19.17	1.00	0.04	0.00
TRIANGLE- TIREWORLD																		
10	4.0	1.17	11.04	0.72	0.11	0.28	4.0	1.67	15.91	0.67	0.11	0.33	4.0	1.00	106.56	0.78	0.26	0.22
30		2.83	10.55	1.00	0.03	0.00		3.83	15.87	0.72	0.07	0.28		2.33	104.83	0.94	0.17	0.06
50		4.50	10.55	0.94	0.01	0.06		6.17	16.13	0.89	0.03	0.11		3.50	102.41	1.00	0.12	0.00
70		6.17	10.58	1.00	0.00	0.00		8.50	14.87	1.00	0.00	0.00		4.67	102.36	1.00	0.25	0.00
100		8.00	11.89	1.00	0.00	0.00		11.33	15.16	1.00	0.00	0.00		6.00	102.91	1.00	0.00	0.00
ZENO- TRAVEL																		
10	4.0	2.50	174.38	0.89	0.03	0.11	4.0	2.17	144.87	0.83	0.04	0.17	4.0	2.00	175.58	0.89	0.14	0.11
30		6.50	167.33	0.89	0.03	0.11		6.17	140.15	0.94	0.01	0.06		5.33	171.27	1.00	0.10	0.00
50		10.17	164.68	0.83	0.04	0.17		9.83	142.72	0.94	0.01	0.06		8.67	167.39	0.94	0.04	0.06
70		14.33	161.67	0.89	0.03	0.11		13.83	138.64	1.00	0.00	0.00		12.67	159.45	1.00	0.00	0.00
100		20.00	161.05	0.83	0.04	0.17		19.50	137.53	1.00	0.00	0.00		17.33	154.69	1.00	0.00	0.00

of the time in the observation sequence for this recognition example.

Figure 7 aggregates the average recognition results of all the six datasets for the *Ranked First* metric as a histogram, by considering full observation sequences that represent executions (sequences of executed actions) of strong-cyclic

policies that achieves the actual intended goal φ^* . The results represent the overall percentage (including the standard deviation – black bars) of how many times the of time that the correct intended goal φ^* has been ranked first over the observations. The average results indicated our approach to be in general accurate to recognize correctly the *temporal order*

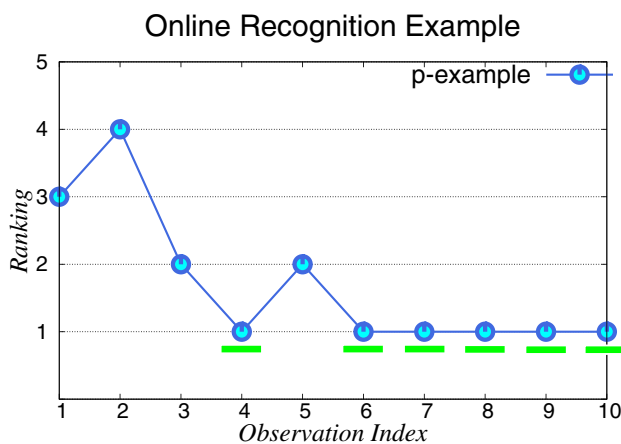


Fig. 6 Online Recognition example

of the facts in the goals in the *Online* recognition setting, yielding *Ranked First* percentage values greater than 58%.

Figures 8, 9, 10, 11, 10, 12, and 13 shows the *Online* recognition results separately for all six domains models and the different types of temporally extended goals. By analyzing the *Online* recognition results more closely, one can see that our approach converges to rank the correct goal as the *top-1* mostly after a few observations. This means that it is commonly hard to disambiguate among the goals at the beginning of the execution, which, in turn, directly affects the overall *Ranked First* percentage values (as shown in Fig. 7). Here, our approach struggles to disambiguate and recognize correctly the intended goal for some recognition problems and some types of temporal formulas. Namely, our approach has struggled to disambiguate when dealing with LTL_f Eventuality goals in BLOCKS- WORLD (see Fig. 8a), for most temporal extended goals in TIDY- UP (see Fig. 10), and for LTL_f Eventuality goals in ZENO- TRAVEL (see Fig. 13a).

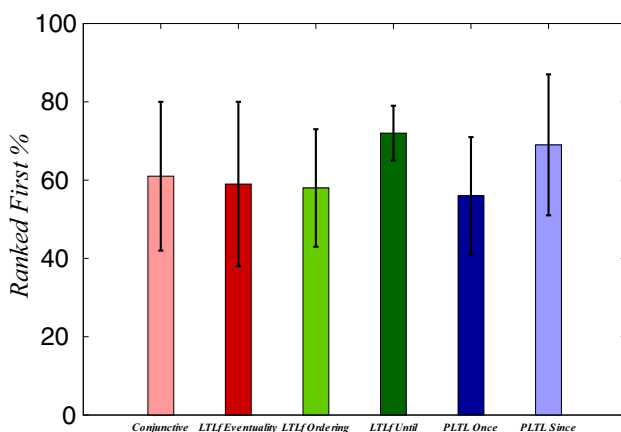


Fig. 7 Online Recognition Histogram

6 Related work and discussion

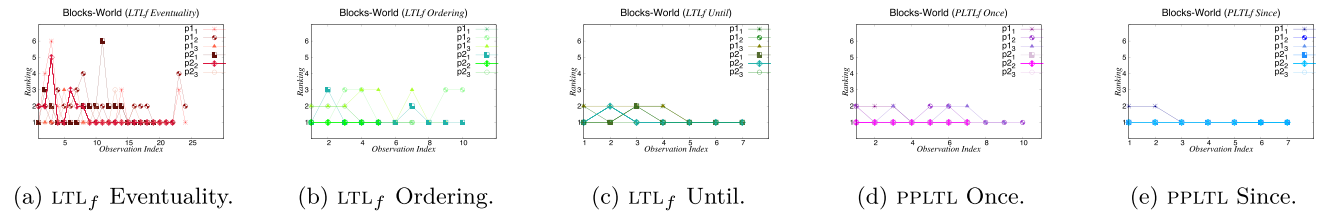
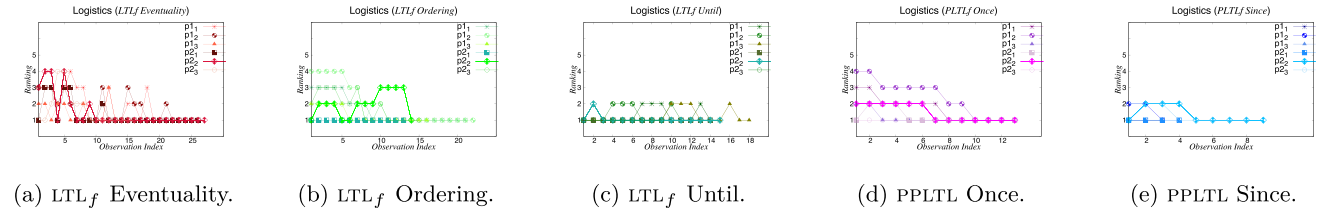
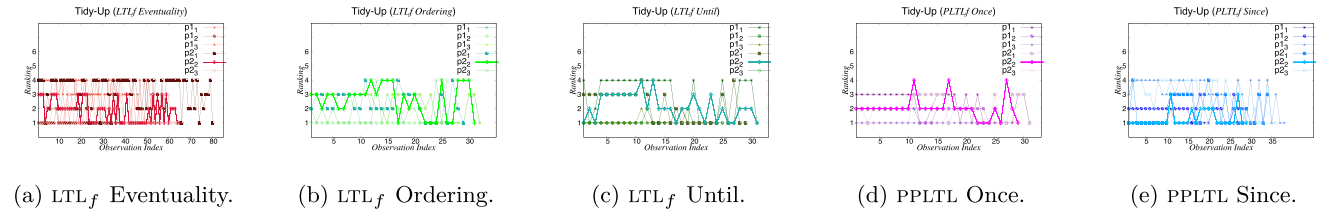
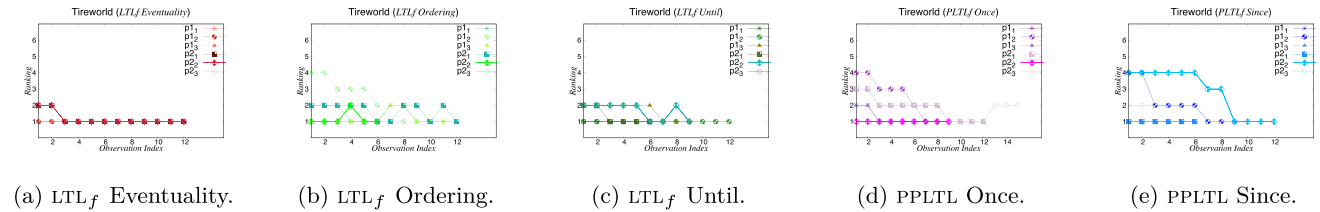
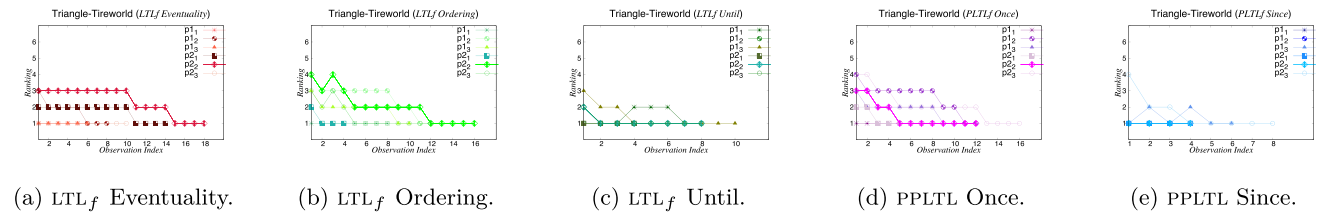
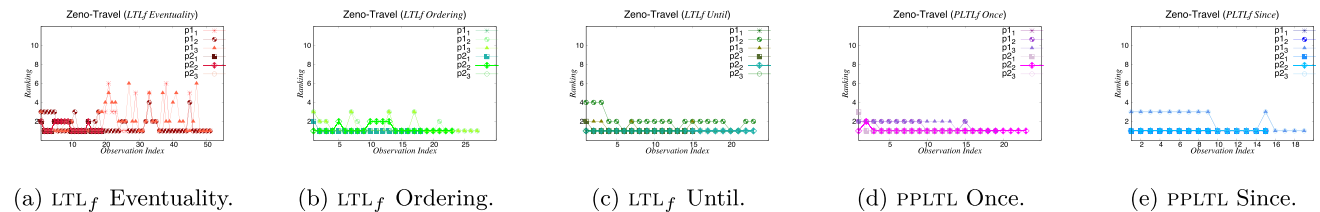
To the best of our knowledge, existing approaches to *Goal and Plan Recognition as Planning* cannot explicitly recognize temporally extended goals in non-deterministic environments. Seminal and recent work on *Goal Recognition as Planning* relies on deterministic planning techniques [31, 34, 37] for recognizing conjunctive goals. By contrast, we propose a novel problem formalization for goal recognition, addressing temporally extended goals (LTL_f or PPLTL goals) in FOND planning domain models. While our probabilistic approach relies on the probabilistic framework of [35], we address the challenge of computing $\mathbb{P}(Obs \mid G)$ in a completely different way.

There exist different techniques to *Goal and Plan Recognition* in the literature, including approaches that rely on plan libraries [4], context-free grammars [19], and Hierarchical Task Network (HTN) [21]. Such approaches rely on hierarchical structures that represent the knowledge of how to achieve the possible goals, and this knowledge can be seen as potential strategies for achieving the set of possible goals. Note that the temporal constraints of temporally extended goals can be adapted and translated to such hierarchical knowledge. For instance, context-free grammars are expressive enough to encode temporally extended goals [11]. LTL_f has the expressive power of the star-free fragment of regular expressions and hence captured by context-free grammars. However, unlike regular expressions, LTL_f uses negation and conjunction liberally, and the translation to regular expression is computationally costly. Note, being equally expressive is not a meaningful indication of the complexity of transforming one formalism into another. [17] show that, while LTL_f and PPLTL have the same expressive power, the best translation techniques known are worst-case 3EXP-TIME.

As far as we know, there are no encodings of LTL_f -like specification languages into HTN, and its difficulty is unclear. Nevertheless, combining HTN and LTL_f could be interesting for further study. HTN techniques focus on the knowledge about the decomposition property of traces, whereas LTL_f -like solutions focus on the knowledge about dynamic properties of traces, similar to what is done in verification settings. Most recently, [7] develop a novel Pure-Past Linear Temporal Logic PDDL encoding for planning in the *Classical Planning* setting.

7 Conclusions

This article introduced a novel problem formalization for recognizing *temporally extended goals*, specified in either LTL_f


Fig. 8 Online recognition ranking over the observations for BLOCKS- WORLD

Fig. 9 Online recognition ranking over the observations for LOGISTICS

Fig. 10 Online recognition ranking over the observations for TIDY- UP

Fig. 11 Online recognition ranking over the observations for TIREWORLD

Fig. 12 Online recognition ranking over the observations for TRIANGLE- TIREWORLD

Fig. 13 Online Recognition ranking over the observations for ZENO- TRAVEL

or PPLTL, in FOND planning domain models. It also developed a novel probabilistic framework for goal recognition in such settings, and implemented a compilation of temporally extended goals that allows us to reduce the problem of FOND planning for LTL_f/PPLTL goals to standard FOND planning. Our experiments have shown that our recognition approach yields high accuracy for recognizing temporally extended goals (LTL_f/PPLTL) in different settings (*Keyhole Offline* and *Online* recognition) at several levels of observability.

As future work, we intend to extend and adapt our recognition approach for being able to deal with spurious (noisy) observations, and recognize not only the temporal extended goals but also anticipate the policy that the agent is executing to achieve its goals.

Acknowledgements This work has been partially supported by the ERC-ADGWhiteMech (No. 834228), the EU ICT-48 2020 project TAILOR (No. 952215), the PRIN project RIPER (No. 20203FFYLK), and the PNRR MUR project FAIR (No. PE0000013).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aineto D, Jimenez S, Onaindia E (2021) Generalized temporal inference via planning. In: KR
- Aminof B, De Giacomo G, Rubin S (2020) Stochastic fairness and language-theoretic fairness in planning in nondeterministic domains. In: ICAPS
- Armentano G, Amandi A (2007) Plan recognition for interface agents. *Artif Intell Rev* 28(2):131–162
- Avrahami-Zilberbrand D, Kaminka GA (2005) Fast and complete symbolic plan recognition. In: IJCAI
- Bacchus F, Kabanza F (1998) Planning for temporally extended goals. *Ann Math Artif Intell* 22(1–2)
- Baier JA, McIlraith SA (2006) Planning with temporally extended goals using heuristic search. In: AAAI
- Bonassi L, De Giacomo G, Favorito M, Fuggitti F, Gerevini A, Scala E (2023) Planning for temporally extended goals in pure-past linear temporal logic. In: ICAPS
- Bryce D, Buffet O (2008) 6th International planning competition: Uncertainty Part. International Planning Competition (IPC)
- Camacho A, Triantafyllou E, Muise C, Baier J, McIlraith S (2017) Non-deterministic planning with temporally extended goals: LTL over finite and infinite traces. In: AAAI
- Camacho A, Baier J, Muise C, McIlraith S (2018) Finite LTL synthesis as planning. In: ICAPS
- Chiari M, Bergamaschi D, Mandrioli D, Pradella M (2020) Linear temporal logics for structured context-free languages. In: ICTCS
- Cimatti A, Giunchiglia E, Giunchiglia F, Traverso P (1997) Planning via model checking: A decision procedure for ar. In: ECP
- Cimatti A, Pistore M, Roveri M, Traverso P (2003) Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence* 147(1–2)
- De Giacomo G, Rubin S (2018) Automata-theoretic foundations of FOND planning for LTL_f and LDL_f goals. In: IJCAI
- De Giacomo G, Vardi M (2013) Linear temporal logic and linear dynamic logic on finite traces. In: IJCAI
- De Giacomo G, Vardi M (2015) Synthesis for LTL and LDL on finite traces. In: IJCAI
- De Giacomo G, Di Stasio A, Fuggitti F, Rubin S (2020) Pure-past linear temporal and dynamic logic on finite traces. In: IJCAI
- Geffner H, Bonet B (2013) A concise introduction to models and methods for automated planning. Morgan & Claypool Publishers
- Geib CW, Goldman RP (2009) A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence* 173(11)
- Henriksen J, Jensen J, Jørgensen M, Klarlund N, Paige B, Rauhe T, Sandholm A (1995) Mona: Monadic second-order logic in practice. In: TACAS
- Höller D, Behnke G, Bercher P, Biundo S (2018) Plan and goal recognition as htn planning. In: International conference on tools with artificial intelligence
- Kaminka GA, Vered M, Agmon N (2018) Plan recognition in continuous domains. In: AAAI
- Mattmüller R, Ortlieb M, Helmert M, Bercher P (2010) Pattern database heuristics for fully observable nondeterministic planning. In: ICAPS
- McDermott D, Ghallab M, Howe A, Knoblock C, Ram A, Veloso M, Weld D, Wilkins D (1998) PDDL – The planning domain definition language. In: AIPS
- Muise C, McIlraith SA, Beck JC (2012) Improved non-deterministic planning by exploiting state relevance. In: ICAPS
- Nebel B, Dornhege C, Hertle A (2013) How much does a household robot need to know in order to tidy up your home? In: AAAI Workshop on robotics
- Patrizi F, Lipovetzky N, Giacomo GD, Geffner H (2011) Computing infinite plans for LTL goals using a classical planner. In: IJCAI
- Patrizi F, Lipovetzky N, Geffner H (2013) Fair LTL synthesis for non-deterministic systems using strong cyclic planners. In: IJCAI
- Pereira RF, Pereira AG, Meneguzzi F (2019) Landmark-enhanced heuristics for goal recognition in incomplete domain models. In: ICAPS
- Pereira RF, Vered M, Meneguzzi F, Ramírez M (2019) Online probabilistic goal recognition over nominal models. In: IJCAI
- Pereira RF, Oren N, Meneguzzi F (2020) Landmark-based approaches for goal recognition as planning. *Artificial Intelligence* 279
- Pnueli A (1977) The temporal logic of programs. In: FOCS
- Pnueli A, Rosner R (1989) On the synthesis of a reactive module. In: POPL, pp 179–190
- Ramírez M, Geffner H (2009) Plan recognition as planning. In: IJCAI
- Ramírez M, Geffner H (2010) Probabilistic plan recognition using off-the-shelf classical planners. In: AAAI
- Ramírez M, Geffner H (2011) Goal recognition over pomdps: inferring the intention of a pomdp agent. In: IJCAI
- Sohrabi S, Riabov AV, Udrea O (2016) Plan recognition as planning revisited. In: IJCAI
- Torres J, Baier JA (2015) Polynomial-time reformulations of LTL temporally extended goals into final-state goals. In: IJCAI
- Vered M, Kaminka GA, Biham S (2016) Online goal recognition through mirroring: Humans and agents. In: Advances in cognitive systems, vol 4

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.


Ramon Fraga Pereira is a researcher in the field of Artificial Intelligence, particularly in the area of Automated Planning. His research interests include Goal and Plan Recognition and Heuristic Search for Sequential Decision-Making. He is currently a Lecturer in Symbolic AI and Autonomous Systems in the Department of Computer Science at the University of Manchester, England, UK. From April 2020 to December 2022, he was a Research Associate at the Dipartimento di Ingegneria Informatica, Automatica e Gestionale (DIAG) at Sapienza Università di Roma "La Sapienza", working with Prof. Dr. Giuseppe De Giacomo on his Advanced ERC project WhiteMech. He obtained his Ph.D. degree in 2020, at Pontifical Catholic University of Rio Grande do Sul (PUCRS), Porto Alegre, Brazil, under the supervision of Prof. Dr. Felipe Meneguzzi (PUCRS) and Dr. Miquel Ramírez (University of Melbourne). During his Ph.D. (from June 2018 to April 2019), he was a Ph.D. Intern in the School of Computing and Information at the University of Melbourne, under the supervision of Dr. Miquel Ramírez. In 2020, he was recognised as having the best Ph.D. thesis in Artificial Intelligence in Brazil, CTDIAC at 9th the Brazilian Conference on Intelligent Systems (BRACIS).

Francesco Fuggitti is a Research Scientist at IBM Research AI in the AI Composition Lab, Cambridge (MA). His research interests include sequential decision-making, formal methods for Artificial Intelligence, conversational intelligence, natural language understanding, process management and automation, and large language models. Francesco received a double Ph.D. from Sapienza University in Rome and York University in Toronto, under the supervision of Professors Giuseppe De Giacomo and Yves Lespérance. Francesco's Ph.D. works received international recognitions, including the Best Student Paper Award and Best System Demonstration Award Runner-Up at ICAPS 2023.

Felipe Meneguzzi is a researcher on Automated Planning, Goal and Plan recognition, Multiagent Systems, BDI Agents, and Machine Learning. He currently holds a Chair of Computing Science at the University of Aberdeen, and is a part of the Agents at Aberdeen research group. He holds a Bridges Professorship at Pontifícia Universidade Católica do Rio Grande do Sul. He is a Senior Member of the ACM and of AAAI. He is currently a councillor in the Executive Council of the AAAI, and a board member of the Special Committee on Artificial Intelligence for the Brazilian Computer Society. As a UK-based researcher, he is a member of the UKRI Talent Panel College, and of the EPSRC Peer Review College. In Brazil, he held CNPq's highly productive researcher fellowship.

Professor Giuseppe De Giacomo is a Governing Body Fellow, Green Templeton College and a Professor of Computer Science, Department of Computer Science, University of Oxford. He was previously professor at the Department of Computer, Control and Management Engineering of the University of Rome "La Sapienza". His research activity has concerned theoretical, methodological, and practical aspects in different areas of artificial intelligence and computer science, most prominently Knowledge Representation, Reasoning about Actions, Generalised Planning, Autonomous Agents, Service Composition, Business Process Modeling, and Data Management and Integration. He is an Association for the Advancement of Artificial Intelligence (AAAI) Fellow, Association for Computing Machinery (ACM) Fellow, and European Association for Artificial Intelligence: EurAI Fellow. He has got an European Research Council (ERC) Advanced Grant for the project WhiteMech: White-box Self Programming Mechanisms (2019-2024). He was the Program Chair of the European Conference on Artificial Intelligence (ECAI) 2020. He is on the Board of EurAI.

Authors and Affiliations

Ramon Fraga Pereira¹  · Francesco Fuggitti^{2,3,4} · Felipe Meneguzzi⁵ · Giuseppe De Giacomo^{3,6}

¹ University of Manchester, Manchester, UK

² IBM Research AI, Cambridge, USA

³ Sapienza University, Rome, Italy

⁴ York University, Toronto, Canada

⁵ University of Aberdeen, Scotland, UK

⁶ University of Oxford, England, UK