A Survey on Goal Recognition as Planning

Felipe Meneguzzi and Ramon Fraga Pereira

1Pontifical Catholic University of Rio Grande do Sul, Brazil
2Sapienza University of Rome, Italy
felipe.meneguzzi@pucrs.br, pereira@diag.uniroma1.it

Abstract

Goal Recognition is the task of inferring an agent’s goal, from a set of hypotheses, given a model of the environment dynamic, and a sequence of observations of such agent’s behavior. While research on this problem gathered momentum as an offshoot of plan recognition, recent research has established it as a major subject of research on its own, leading to numerous new approaches that both expand the expressivity of domains in which to perform goal recognition and substantial advances to the state-of-the-art on established domain types. In this survey, we focus on the advances to goal recognition achieved in the last decade, categorizing the resulting techniques and identifying a number of opportunities for further breakthrough research.

1 Introduction

Goal Recognition is a task related to automated planning, where an agent employs abductive reasoning to infer the most likely desired goal from a series of observations of the observed agent’s plan instead of deducing a plan from an initial state towards a goal using some kind of domain theory. While goal recognition is related to the problem of Plan Recognition [Mirskey et al., 2021], which consists of trying to infer the actual plan being carried out by the observed agent, this survey focuses squarely on goal recognition. The task of goal recognition has a number of potential and actual applications, including assisting the handicapped [Geib, 2002], activities of daily living (e.g., cooking) [Granada et al., 2017], workplace safety [Inam et al., 2018], smart home [Hegde and Kenchannavar, 2019], among others [Singh et al., 2020; Wayllace et al., 2020a]. As research on goal recognition evolves into ever more complex domain models and better approaches, we expect the state-of-the-art to advance substantially and new application domains to be broken into.

Thus, we survey advances on goal recognition achieved primarily in the last decade, defining a common formal framework in Section 2, and exploring key aspects that distinguish current approaches. We formalize types of domain models that modern goal recognition approaches process in Section 3, followed by an analysis of the effect of different types of observations in Section 4. Section 5 looks into the assumptions each recognition approach makes about the agent being observed, especially regarding the awareness of the agent to the observer and how optimal the observed agent is. We evaluate the internal mechanism through which goal recognition approaches infer goals from observations in Section 6. Finally, we discuss various problems related to goal recognition in Section 7, and organize the approaches surveyed in this paper, pointing how future work can solve many of the limitations in the current state-of-the-art in Section 8.

2 Goal Recognition as Planning

We lay out the formal foundations of the problem we survey using a top-down approach and frame goal recognition in the context of automated planning, outlining the basic environment model, which we refine throughout the paper to match the specific approaches we discuss. We then formalize the task of goal recognition itself based on a planning model, which we also refine throughout the survey.

2.1 Automated Planning

Planning aims to select what an agent does next, given a model of the environment specifying how actions and sensors work, a current situation, and what is the goal to be achieved [Geffner and Bonet, 2013]. Such problem can be seen as a directed graph in which nodes represent states, edges represent the transition between states (caused by applying actions), and the solution is a path (i.e., plan) between two particular nodes (i.e., initial state and goal state) in this directed graph. Such graph is induced by a planning task following Definition 1.

Definition 1 (Planning Task) A planning task \( \Pi = (\mathcal{Z}, s_0, G) \) is a tuple composed of a domain definition \( \mathcal{Z} \), an initial state \( s_0 \), and a goal state specification \( G \). A solution to a planning task is a plan or policy \( \pi \) that reaches a goal state \( G \) starting from the initial state \( s_0 \) by following the transitions defined in the domain definition \( \mathcal{Z} \).

2.2 Goal Recognition

Goal Recognition is the task of recognizing which goal an agent aims to achieve by observing its interactions in an environment [Sukthankar et al., 2014, Chapter 1, Page 3]. Such observed interactions (i.e., observations) constitute the evidence to recognize goals, and can be executed actions in...
the environment (e.g., a simple movement, cook, drive), and changing properties in an environment (e.g., at home, at work, resting). In this survey, we consider a goal recognition task by following the original problem defined by Ramírez and Geffner [2009]. As we formalize in Definition 2, a goal recognition task follows the same structure of a planning task (Definition 1), but instead of having a single goal condition, it has a set of goal hypotheses, and a sequence of observations induced by the actions executed by the observed agent. Definition 2 does not detail what a solution to a goal recognition problem is, as different approaches provide different solution concepts, which we provide in Section 6.2.

**Definition 2 (Goal Recognition Task)** A goal recognition task \( \Pi^G \) is a tuple composed of a domain definition \( \Xi \), an initial state \( s_0 \), a set of goal hypotheses \( G \), and a sequence of observations \( \Omega \).

Most goal recognition tasks assume there is at least one correct intended goal \( G^* \in G \) among the hypotheses. However, as we see in Section 6.1, how exactly to present such solutions varies with the specific approach. Our definitions so far are vague into exactly what constitutes a domain definition, and what form the observations take. This is a deliberate presentation choice, as goal recognition approaches assume a variety of different models and observation types. Thus, we use \( \Xi \) to refer to any type of domain model, and pin down these types in Section 3. Likewise, we use \( \Omega \) to refer to any observation sequence so we can define recognition problems independently of the observations, which we detail in Section 4. Example 1 illustrates a goal recognition task.

**Example 1** Figure 1 illustrates a navigation environment where a robot can move and pick up objects to achieve its goals. The goal recognition task presented in the example is composed of an initial state \( s_0 \) (e.g., robot position), a set of goal hypotheses \( G = \{G_0, G_1, G_2\} \), and a sequence of observations \( \Omega \) (e.g., a sequence of movements, represented by the dashed blue arrows). The domain definition \( \Xi \) can vary according to the domain type used to formalize the environment dynamics, e.g., a deterministic discrete domain model, a stochastic domain model, among others. For this example, we consider that \( \Xi \) is a deterministic discrete domain model. A goal state specification \( G \) for this example can be defined as the robot be at a certain location holding an object (e.g., \( G_1 \) represents the robot holding the ball in the upper right corner of the environment). Thus, based on the goal recognition task depicted in Figure 1, it is possible to infer that, most likely, the robot aims to achieve the goal condition \( G_2 \) (e.g., holding the box in the bottom right corner of the environment).

3 Domain Model Types

To the best of our knowledge, the first practical goal recognition approach using a domain model recognizable by current approaches goes back to Lesh and Etzioni [1995]. This approach, instead of abstracting plans to strings from a grammar, uses a state transition system induced by a planning domain much like that of Definition 1. Much of the early plan and goal recognition approaches focused treating the problem as one of parsing, assuming the domain knowledge includes not only the state transition function, but also production rules restricting the space of possible plans [Geib and Steedman, 2007]. In effect, such planning domains induce graphs \( \langle S, A \rangle \) via their transition function \( \gamma \), with vertices \( S \) and edges \( A \) as in [Hong, 2001]. Definition 1 abstracts the specifics of the domain definition and what constitutes a state and a transition system. Indeed, \( \Xi \) induces a number of elements of a planning task, including a state space \( S \), an action space \( A \), and a transition function \( \gamma \). Throughout this survey we refine domain model types defining what exactly states, actions, and transition systems can be.

3.1 Classical Planning Domain Models

The most common domain model in goal recognition research consists of the STRIPS fragment of PDDL [Fox and Long, 2003a], as used by the seminal work of Ramírez and Geffner [2009] and much subsequent work [Martin et al., 2015; Sohrabi et al., 2016; Pereira et al., 2017; Pereira et al., 2020]. In Classical Planning, the state space is often represented as a set of propositional facts \( F \) (i.e., instantiated predicates) denoting what is true in the environment, or, alternatively, as multivalued variables in a finite domain representation [Helmert, 2009]. Regardless of the specific representation, a state comprises a set of variables \( V \) with a finite domain and the state space \( S \) is finite and well-defined. Likewise, in this setting, the action space \( A \) comprises STRIPS-style actions of the form \( a = \langle \text{name}(a), \text{pre}(a), \text{post}(a), \text{cost}(a) \rangle \) with conjunctive preconditions \( \text{pre}(a) \) and postconditions \( \text{post}(a) = \text{post}^+(a) \cup \text{post}^-(a) \) comprising both positive \( \text{post}^+(a) \) and negated facts \( \text{post}^-(a) \). Such actions induce a transition function \( \gamma : S \times A \rightarrow S \) such that \( \gamma(s, a) = s' \), where \( s' = (s \cup \text{post}^+(a)) - \text{post}^-(a) \). For example, for the navigation environment depicted in Example 1, a Classical Planning task II comprises a set of instantiated facts, representing the state properties of this navigation environment, where the robot and the objects are situated (e.g., \( \langle \text{at robot loc-0-0}, \text{at box loc-0-5} \rangle \)), and what objects the robot is holding (e.g., \( \text{holding ball} \)), the action space \( A = \langle \text{move}, \text{pick-up} \rangle \) (we omit \( \text{pre} \) and \( \text{post}^+ \) due to lacking of space), \( s_0 = \langle \text{at robot loc-0-0} \text{ at box loc-0-5} \rangle \), and \( G_0 = \langle \langle \text{at robot loc-2-5} \text{ holding ball} \rangle \rangle \).

3.2 Stochastic Domain Models

By contrast, stochastic domains are often modeled in terms of (Partially Observable) Markov Decision Processes (POMDPs and MDPs), as in the approaches of Ramírez

---

\[1\]While space limitations prevent us from providing details of the PDDL format, we refer to Haslum et al. [2019]
and Geffner [2011] and Oh et al. [2011]. In such models, MDPs transition functions $\gamma$ are stochastic and such that $s' \sim \gamma(s, a)$, i.e., subsequent states are sampled following a stochastic function $\mathbb{P}[s' \mid s, a]$ for $s, s' \in S$, $a \in A$.

### 3.3 Continuous Domain Models

Vered and Kaminka [2017a; 2017b] formally define the task of goal recognition by using the formalism of Continuous Motion Planning. In this setting, the state space $S$ of a domain model $\Xi$ is represented in a multi-dimensional Euclidean space, where $S \subseteq \mathbb{R}^n$, such that $n \geq 2$. This type of domain model usually represents environments with two or three dimensions. The action space $A$ is a discrete (possibly infinite) set of actions, encoding a transition function $\gamma$ between states. This transition function $\gamma$ allows transforming one state $(s,r)$ into another $(s',r)$ via paths through the state space (e.g., a non-negative straight-line from $s$ to $s'$), rather than a discrete state, as defined in Classical Planning. We note that other recognition approaches in the literature also adopt this type of domain model, such as [Masters and Sardina, 2017; Vered et al., 2018; Kaminka et al., 2018; Masters and Sardina, 2019a; Masters and Sardina, 2019b]. For the navigation environment depicted in Example 1, a Continuous Motion Planning task II could be defined with 3 dimensions $(x, y, z)$, so the actions space $A = \{\text{move, pick-up}\}$ would be performed over continuous values in $x, y, z$. Stochastic domains driven by reward functions [Sutton and Barto, 2018], relatively less emphasis has been given to the automatic learning of Classical Planning models, with few techniques capable of generating the type of lifted PDDL models used by most of the approaches described in this survey. Recent work yielded mechanisms to learn classical planning models from structured [Aïnès et al., 2019b; Suárez-Hernández et al., 2020] and unstructured data [Asai and Fukunaga, 2018].

These advances have led to a number of approaches for goal recognition using automatically derived models. Amado et al. [2018] develop a goal recognition approach that learns PDDL models from unstructured data allowing symbolic approaches [Pereira et al., 2017] to quickly solve problems encoded in images. Similarly, Pereira et al. [2019a] adapt existing landmark-based goal recognition approaches to deal with incomplete STRIPS domain models [Weber and Bryce, 2011; Nguyen et al., 2017], which assume that the model available to the recognizer might contain imprecisions (possible preconditions and effects) due to imperfect model acquisition. Pereira et al. [2019b] develop a goal recognition approach for continuous control domains that uses an approximate learned transition function and an optimal policy to infer goals by adapting existing symbolic techniques [Vered et al., 2016]. Both approaches achieve a level of accuracy comparable to traditional approaches that rely on authored domains\(^2\), indicating that traditional recognition approaches can be augmented to cope with learned models, even when the models themselves might have flaws from the learning process.

### 3.4 Optimal Control Domain Models

Optimal Control aims to control a dynamical system such that its output follows a desired value, which may be a fixed or changing value [Bertsekas, 2017]. Pereira et al. [2019b] use concepts of Optimal Control for recognizing goals in approximate continuous domain models. To model Optimal Control problems and the range of possible agent behavior, the authors adopt the formalism of Finite-Horizon Optimal Control (FHOC) [Bertsekas, 2017] problems, incorporating and combining some terminologies of Control [Borrelli et al., 2017] and Automated Planning [Geffner and Bonet, 2013] to account for constraints and goal conditions (also referred to as target regions in the Control literature).

In this setting, transitions between states follow a stationary, discrete-time dynamical system $x_{k+1} = \gamma(x_k, u_k, w_k)$, where for each time point $k \in [0, N]$, $x_k$ is the state, $u_k$ is the control input and $w_k$ is a random variable with a probability distribution that does not depend on past $w_j, j < k$. Stages $x_k \in S$, controls $u_k \in U$, and disturbances $w_k \in W$ are required to be part of $\mathbb{R}^n$. For extracting trajectories, agents seek to transform initial states $x_0$ into states $x_N$ with specific properties. These properties are given as logical formulas over the components of states $x_k$, and the set of states $S_N \subseteq S$ are those where the desired property $G$, or goal, holds. The preferences of observed agents to pursue specific trajectories are accounted for with cost functions of the form $J(x_0) = \mathbb{E}(g(x_N) + \sum_{k=0}^{N-1} g(x_k, u_k, w_k))$, where $g(x_N)$ is the terminal cost, and $g(x_k, u_k, w_k)$ is the stage cost. FHOC problems are defined as an optimization problem whose solutions are trajectories (i.e., policies) that describe the range of possible optimal behaviors of observed agents.

### 3.5 Approximate, Incomplete, and Learned Domain Models

Regardless of the nature of the underlying model, one of the key bottlenecks in planning, and by extension, goal recognition, is the acquisition of a model II from either structured or unstructured data. While research on reinforcement learning has yielded reliable model acquisition techniques for stochastic domains driven by reward functions [Sutton and Barto, 2018], relatively less emphasis has been given to the automatic learning of Classical Planning models, with few techniques capable of generating the type of lifted PDDL models used by most of the approaches described in this survey. Recent work yielded mechanisms to learn classical planning models from structured [Aïnès et al., 2019b; Suárez-Hernández et al., 2020] and unstructured data [Asai and Fukunaga, 2018].

These advances have led to a number of approaches for goal recognition using automatically derived models. Amado et al. [2018] develop a goal recognition approach that learns PDDL models from unstructured data allowing symbolic approaches [Pereira et al., 2017] to quickly solve problems encoded in images. Similarly, Pereira et al. [2019a] adapt existing landmark-based goal recognition approaches to deal with incomplete STRIPS domain models [Weber and Bryce, 2011; Nguyen et al., 2017], which assume that the model available to the recognizer might contain imprecisions (possible preconditions and effects) due to imperfect model acquisition. Pereira et al. [2019b] develop a goal recognition approach for continuous control domains that uses an approximate learned transition function and an optimal policy to infer goals by adapting existing symbolic techniques [Vered et al., 2016]. Both approaches achieve a level of accuracy comparable to traditional approaches that rely on authored domains\(^2\), indicating that traditional recognition approaches can be augmented to cope with learned models, even when the models themselves might have flaws from the learning process.

### 4 Observation Types

Observations constitute the key piece of evidence for agent behavior in goal recognition problems. In practice, observations are an indirect projection of an observed agent’s behavior, and can be thought of as whatever the sensing capabilities of the agent performing goal recognition can provide. As we see in Section 6.2, the model we assume drives agent behavior influences the nature of how observations are generated. Such dependency on the model leads us to use Definition 3 as an intermediary structure between an agent’s behavior model and the generation of observations.

**Definition 3 (Trajectories)** Let $\pi$ be a solution to a planning task II, $\pi$ is a trajectory induced by $\pi$. For Classical Planning tasks, $\pi$ is a sequential plan $\pi$; for MDPs and stochastic

\(^2\)We refer to authored domains as domain models written by a human using a specification language like PDDL.
problems $\pi$ is sequence of transitions from $s_0$ to $G$ following policy $\pi$ while complying with the transition function; and for Continuous Planning or Control tasks, $\pi$ is a sequence of poses or configurations (e.g., coordinates $(x, y, z)$ and a velocity) between which there are infinite intermediary poses.

Having defined trajectories, we can formally define how actual observations are generated in Definition 4.

**Definition 4 (Observations)** Let $\pi = \langle \bar{a}_1, \ldots, \bar{a}_n \rangle$ be a trajectory for the planning task $\Pi = (\mathcal{Z}, s_0, G)$. An action projection function $tp(s, \bar{a}) : S \times A \rightarrow A$ is a function that maps actions to sequences of zero or more observations. An observation sequence generation function $op(s_0, \pi)$ is a function mapping a trajectory $\pi$ into an observation sequence:

$$op(s_0, \pi) = \begin{cases} \emptyset & \text{if } \pi = \emptyset \\ \{tp(s_0, a_1) \circ op(\langle \bar{a}_2, \ldots, \bar{a}_n \rangle) & \text{if } \pi = \langle \bar{a}_1, \ldots, \bar{a}_n \rangle \end{cases}$$

Then, a sequence of observations $\Omega$ is a sequence projected from $\pi$ to observation state $\bar{A}$ maintaining their order.

The sensors available to the observer imposes limitations on types of observations. While most approaches construe observations as action descriptions of some sort, this is not necessarily the most realistic abstraction of how sensor data may be supplied to a goal recognizer. For instance, modern computer vision approaches to object detection [Wang et al., 2020] may provide a more reliable representation of the current status of objects in an image than action identification. Indeed, recent work on goal recognition as planning [Sohrabi et al., 2016] explicitly handles observations as states, the difference of which we formally define as follows.

**Definition 5 (Action and State Observations)** Let $\pi = \langle \bar{a}_1, \ldots, \bar{a}_n \rangle$ be a trajectory for the planning task $\Pi$ and $tp(\cdot, \cdot)$ be an action projection. A sequence of observations $\Omega_\pi$ is a sequence of actions from $\pi$ if $tp(\cdot, \cdot) : S \times A \rightarrow A$, i.e., if $\bar{A} = A$. Alternatively, a sequence of state observations $\Omega_S$ is a sequence of states from $S$ if $tp(\cdot, \cdot) : S \times A \rightarrow S$, i.e., if $\bar{A} = S$. We denote the observation corresponding to action $a_i$ as $a_i^\pi$ and that corresponding to state $s_i$ as $s_i^\Omega$.

For the navigation environment depicted in Example 1, action observations in the Classical Planning setting could be $\langle \bar{a}_0 \rangle = \langle \text{move loc-0-0 loc-0-1} \rangle, \bar{a}_1 = \langle \text{move loc-0-1 loc-0-2} \rangle$, and state observations in Continuous Motion Planning setting could be $\langle s_0 \rangle = \langle x(r) = 0.28, y(r) = 1.44, z(r) = 0.93, \theta(r) = 45^\circ, s_1 \rangle = \langle x(r) = 0.55, y(r) = 3.73, z(r) = 1.12, \theta(r) = 90^\circ \rangle$.

Given limitations in the sensing capability of the recognizer, observations may contain flaws, from missing to outright wrong/noisy observations, formalized in Definition 6.

**Definition 6 (Missing and Noisy Observations)** Let $\Pi$ be a planning task, $\pi$ be a valid plan for $\Pi$ and $\Omega$ be an observation sequence induced by an observation generation function $op(s_0, \cdot)$ with an action projection function $tp(\cdot, \cdot)$. An observation sequence $\Omega$ **misses** observations (is a partial observation sequence) with respect to the plan $\pi$ if the $tp(\cdot, \cdot)$ function contains a mapping $a_i \mapsto ()$ for some action $a_i$, i.e., it maps one or more actions into the empty sequence. An observation sequence $\Omega$ contains **noisy** observations with respect to the plan $\pi$ if the $tp(s_{i-1}, a_i)$ function maps $a_i$ into a non-empty sequence containing either one or more: (i) actions $a_j \neq a_i$ (for action observations); or (ii) states $s_i \not\in \gamma(s_{i-1}, a_i)$.

Indeed, most early approaches deal with missing observations implicitly by imposing a sequential constraint on the occurrence of elements of $\Omega$ in the plans considered plausible for a goal hypothesis $G$ [Ramírez and Geffner, 2009], or ignoring most observations focusing on the overlap with necessary conditions [Pereira et al., 2017]. Most approaches also deal with noisy observations either implicitly [Ramírez and Geffner, 2010; Pereira et al., 2020], or not at all [Ramírez and Geffner, 2009]. Sohrabi et al. [2016], by contrast compiles $\Pi$ into a single planning task with special predicates representing compliance with observations and achievement of goal hypotheses, as well as additional actions to achieve such predicates. This compilation ensures that goals whose plans have the least cost while complying with the most observations have higher probability. An **optimal** sequence of observations is extracted from an optimal plan and a sub-optimal sequence of observations from extracted of a sub-optimal plan.

## 5 Agent Assumptions

There are two key types of assumption regarding the observed agent that most approaches take into account during the recognition process: **awareness**, and **optimality**.

### 5.1 Awareness

The **awareness** assumption determines whether the agent is aware of it being monitored and what the attitude of the agent is towards the observer. Research on goal recognition characterize such awareness during the recognition process as follows [Armentano and Amandi, 2007; Sukthankar et al., 2014]. **Intended Recognition** is the recognition process in which the observed agent is aware of the process of recognition. Therefore, in this type of recognition process the observed agent usually cooperates with the process by notifying the recognizer about its interactions in the environment. **Obscured Recognition**, by contrast, is such that the observed agent is aware of the process of recognition and obstructs purposely the process, so the agent intentionally does not cooperate with the recognition process. **Keyhole Recognition** is a recognition process in which the observed agent is unaware of the process of recognition, namely, the interactions performed by the observed agent are partially observable inputs to the recognition process. This type of assumption on recognition process is the most common, since it allows the recognizer to ignore any process of interpretation of the actions by the observed agent as either adversarial or trying to be cooperative. Indeed, using such minimal level assumptions has led to many of the current state-of-the-art recognition algorithms to be vulnerable to either changes in target agent behavior or outright misdirection, especially when taking into consideration inferred agent preferences [Masters et al., 2021].

### 5.2 Optimality

We now examine what is likely to be the most common assumption among goal recognition models, which is that the
agent under observation either behaves optimally (i.e., perfectly rationally), or at least approximately optimally. The reason for this assumption is grounded on the fact that most early approaches to goal recognition, and indeed many more recent ones, carry out recognition by generating valid plans for all goal hypotheses \( G \). Since it is possible for planning problems to have infinitely many arbitrary plans that achieve \( G \) (e.g., by introducing irrelevant actions), methods that filter plausible goals by generating plans must assume optimality to bound the computation of such plans.

By contrast, methods that yield a probability distribution for \( P[G \mid \Omega] \) must have a criterion to assign probabilities to goal hypotheses, which often follows the assumption that agents under observation are approximately optimal and make decision that agents under observation are approximately optimal. The conditional probability of observations given a goal is directly related to the probability of the observed agent choosing a plan \( \pi \) for a particular goal \( G \), which leads to the most common assumption for plan preference, which is that observed agents are approximately optimal and prefer close to optimal, and thus \( P[G \mid \Omega] \propto (\text{cost}(\Omega) - \text{cost}(\pi^*)) \). Sohrabi et al. [2016] compute the probability \( P[\pi \mid G] \) of the top-k plans \( \pi \) consistent with \( G \) while complying with observations \( \Omega \) as a proxy for \( P[\Omega \mid G] \). Masters and Sardina [2017; 2019a] reformulate the probabilistic interpretation of Ramírez and Geffner [2010] in the context of path-planning, and show that a single-observation recognition yields similar results in less than half of the recognition time. Martín et al. [2015] develop a probabilistic recognition approach that relies on planning graphs. In contrast, Pereira et al. [2020] rank hypotheses \( G \) following a heuristic based on the number of landmarks inferred from observations, returning a set of goals within a \( \theta \) threshold of the highest ranking hypothesis. While probabilistic approaches notionally use a prior probability \( P[G] \) indicating agent preferences over goals, virtually no approach actually uses such prior, and instead assumes a uniform prior over the goals, leading to potential vulnerability to deception [Masters et al., 2021].

6 Goal Recognition Approaches

We now explore approaches in terms of the underlying mechanisms to infer the correct goal, the algorithms underpinning each approach, and how they process observations.

6.1 Filtering vs. Ranking and Probabilities

One of the key assumptions of goal recognition tasks is that there is a single goal \( G^* \) among the goal hypotheses \( G \). Nevertheless, most realistic goal recognition problems impose constraints on the quality of the observations and the level of observability afforded by the recognizer. This means that inferring goals can be fraught with uncertainty when observations suffer from flaws such as noisy and missing observations. Thus, while one may consider the problem of goal recognition as a filtering process aimed at providing all goal hypotheses \( G \) consistent with \( \Omega \) [Ramírez and Geffner, 2009], such set of goals may be empty when \( \Omega \) is too noisy, or encompass all hypothesis space \( G \) when \( \Omega \) is too small. In such cases, it is useful to infer a set of potentially correct goals, either ranked by some preference relation, or, ideally, a probability distribution \( P[G \mid \Omega] \) for a particular goal \( G \), which leads to the most common assumption for plan preference, which is that observed agents are approximately optimal and prefer close to optimal, and thus \( P[G \mid \Omega] \propto (\text{cost}(\Omega) - \text{cost}(\pi^*)) \). Sohrabi et al. [2016] compute the probability \( P[\pi \mid G] \) of the top-k plans \( \pi \) consistent with \( G \) while complying with observations \( \Omega \) as a proxy for \( P[\Omega \mid G] \). Masters and Sardina [2017; 2019a] reformulate the probabilistic interpretation of Ramírez and Geffner [2010] in the context of path-planning, and show that a single-observation recognition yields similar results in less than half of the recognition time. Martín et al. [2015] develop a probabilistic recognition approach that relies on planning graphs. In contrast, Pereira et al. [2020] rank hypotheses \( G \) following a heuristic based on the number of landmarks inferred from observations, returning a set of goals within a \( \theta \) threshold of the highest ranking hypothesis. While probabilistic approaches notionally use a prior probability \( P[G] \) indicating agent preferences over goals, virtually no approach actually uses such prior, and instead assumes a uniform prior over the goals, leading to potential vulnerability to deception [Masters et al., 2021].

6.2 Search vs. Heuristic Computation

While seminal goal recognition approaches rely on various graph analysis algorithms [Lesh and Etzioni, 1995; Hong, 2001], the approaches that have dominated the state-of-the-art in recent years can be broadly divided into two categories.

The first category relies on compilations of the goal recognition task into Classical Planning tasks, running the search procedure of a planner once or more and comparing the resulting plans to extract \( G^* \) [Ramírez and Geffner, 2010; Sohrabi et al., 2016]. The second category uses the structure of planning tasks derived from \( \Pi_2^3 \) by adapting planning techniques to compute \( G^* \) either through the computation of the heuristic or by comparing heuristic values computed for each goal hypothesis. Typical examples of the first category include Ramírez and Geffner [2009] and Sohrabi et al. [2016]. Ramírez and Geffner [2009] generate a planning task \( \Pi_2 \) from \( \Pi_3^3 \) by compiling away the observations such that when a plan tries to solve \( \Pi_2 \), it must generate plans compatible with the observations \( \Omega \). By contrast, typical examples of the second category include the approaches developed by Pattison and Long [2010], Martín et al. [2015], Pereira et al. [2017], Vered and Kaminka [2017b], and Santos [2021]. Pattison and Long [2010] propose a probabilistic recognition approach that uses heuristic estimation and domain analysis to determine which goals an agent is pursuing. Martín et al. [2015] develop a heuristic estimation approach that relies on planning graphs. Pereira et al. [2017] compute the set of landmarks [Hoffmann et al., 2004] for each goal hypothesis \( G \subseteq G \) and computing which landmarks have definitely been reached by plans that comply with the observations. Vered and Kaminka [2017b] propose a very effective recognition algorithm.
heuristic for continuous domain models. Recently, Santos et al. [2021] computes a goal ranking using linear programming over operator counting constraints on the problems induced by the goal hypothesis.

### 6.3 Online and Offline

Approaches to goal recognition differ in the way that the observations are perceived and revealed from the perspective of the recognizer. Most seminal approaches perform the recognition task offline [Ramírez and Geffner, 2009; Ramírez and Geffner, 2010; Martín et al., 2015; Sohrabi et al., 2016; Pereira et al., 2017; Masters and Sardina, 2019a], in which the observations \( \Omega \) are all revealed at once and up front, before starting the recognition process. The converse way to perform the recognition task is online, in which the observations \( \Omega \) are revealed incrementally and recognition takes place multiple times, as each observation is revealed.

**Online and offline** goal recognition are not only different in the way that the observations are perceived and revealed, but also in the way the approaches are designed to perform the recognition task. **Online** goal recognition approaches are designed to be efficient and deal with incremental observations. Essentially, it is possible to perform **online** goal recognition by repeatedly calling an offline recognition approach for every new revealed observation. However, as proven by Vered and Kaminka in [2017a; 2017b], this is quite inefficient. In particular, Vered and Kaminka [2017a; 2017b] developed an online goal recognition that requires at most \( |G| (|\Omega| + 1) \) calls to a planner, and at best \( |G| \) calls, whereas the seminal offline approach of Ramírez and Geffner [2010] requires \( 2 \times |G| \times |\Omega| \) calls to a planner.

### 7 Related Problems

**Goal Recognition** can be seen as a subproblem of **Plan Recognition**, and it is closely related to a variety of problems in the literature. Baker et al. [2009] propose a probabilistic framework based on Bayesian inverse planning for modeling human action understanding. This probabilistic framework attempts to approximate the principle of rationality, which expects that the observed agents will plan approximately rationally to achieve their goals. Ramírez and Geffner [2009; 2010] borrowed the idea of Baker et al. [2009], and formally define **Plan Recognition as Planning**, claiming that **Plan Recognition** can be defined as **Planning** in reverse. Recent research extends the problem formulation of Ramírez and Geffner for recognizing plans in a variety of domain models, such as continuous domain models [Kaminka et al., 2018] and epistemic planning problems [Shvo et al., 2020]. Alternatively, Aineto et al. [2019a] introduce **Model Recognition as Planning**, a novel recognition task that aims to identify the model that explains a sequence of observations.

Keren et al. [2014; 2020] provide an alternate view of the goal recognition task, introducing a novel approach that modifies the domain model in order to facilitate the goal recognition process, so-called **Goal Recognition Design**. The aim of this task is optimizing the domain design so that recognition approaches can provide inferences with as few observations as possible. Over the past years, a variety of goal recognition design (and derived) approaches have been developed over distinct types of domain models and settings, such as [Keren et al., 2016; Wayllace et al., 2016; Keren et al., 2018; Wayllace et al., 2020b; Shvo and McIlraith, 2020].

Pozanco et al. [2018] develop a Counterplanning approach that relies on techniques from goal recognition and planning to prevent observed agents from achieving their goals. Most recently, Bernardini et al. [2020] develop a set of strategies for **Goal Obfuscation**, in which an actor agent aims to maintain its goal private, without revealing it to an observer.

### 8 Conclusions and Perspectives

We have surveyed advances in goal recognition as planning from the last years, classifying the various aspects that distinguish the approaches, including the types of domains being handled, how observations are handled, what assumptions are being made about the agent under observation, culminating in the algorithms that underpin such approaches. We summarize this classification in Table 1, where **Technique** refers to the underlying approach, **Model** refers to the type of model that describes the tasks (whether authored or learned, and the domain description), **Observation Flaws** refers to the potential flaws in the observations, **Agent Assumption** refers to the assumptions made about the agent, and finally, **Solution** refers to the type of solution provided by the approach.

Despite the substantial progress in the approaches we survey, many research challenges remain. Existing approaches assume pure models in some sense, so models are either continuous or symbolic. Thus, future research should focus

<table>
<thead>
<tr>
<th>Approach</th>
<th>Technique</th>
<th>Model</th>
<th>Obs. Flaws</th>
<th>Agent Assumption</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ran et al. [2009]</td>
<td>Search/Compilation</td>
<td>Author</td>
<td>Missing/Complete</td>
<td>Optimal</td>
<td>Filtering</td>
</tr>
<tr>
<td>Ran et al. [2010]</td>
<td>Search/Compilation</td>
<td>Author</td>
<td>Missing</td>
<td>Optimal</td>
<td>Probability</td>
</tr>
<tr>
<td>Ran et al. [2011]</td>
<td>Search/Compilation</td>
<td>Author</td>
<td>Missing</td>
<td>Optimal</td>
<td>Probability</td>
</tr>
<tr>
<td>Ran et al. [2015]</td>
<td>Heuristic</td>
<td>Author</td>
<td>Missing</td>
<td>Optimal</td>
<td>Ranking</td>
</tr>
<tr>
<td>Sohrabi et al. [2016]</td>
<td>Search/Compilation</td>
<td>Author</td>
<td>Missing</td>
<td>Optimal</td>
<td>Ranking</td>
</tr>
<tr>
<td>Masters and Sardina [2017; 2019a]</td>
<td>Search/Compilation</td>
<td>Author</td>
<td>Missing</td>
<td>Optimal</td>
<td>Ranking</td>
</tr>
<tr>
<td>Martín et al. [2017; 2020]</td>
<td>Heuristic</td>
<td>Author</td>
<td>Missing</td>
<td>None</td>
<td>Ranking</td>
</tr>
<tr>
<td>Martín et al. [2019b]</td>
<td>Heuristic/Search</td>
<td>Author</td>
<td>Missing</td>
<td>None</td>
<td>Ranking</td>
</tr>
<tr>
<td>Masters and Sardina [2019b]</td>
<td>Search/Compilation</td>
<td>Author</td>
<td>Missing</td>
<td>Optimal</td>
<td>Ranking</td>
</tr>
<tr>
<td>Pereira et al. [2018]</td>
<td>Heuristic/Search</td>
<td>Author</td>
<td>Missing</td>
<td>None</td>
<td>Ranking</td>
</tr>
<tr>
<td>Santos et al. [2021]</td>
<td>Heuristic</td>
<td>Author</td>
<td>Missing</td>
<td>None</td>
<td>Ranking</td>
</tr>
</tbody>
</table>

Table 1: Goal Recognition landscape.
on what Ramírez calls interesting domains, i.e., models described in PDDL+ [Fox and Long, 2006], or temporal planning [Fox and Long, 2003b]. Similarly, few approaches deal with multi-agent domains, which we do not survey [Shvo et al., 2017; Argenta et al., 2017; Zhuo, 2019].

The key challenge to current incremental research is comparing existing approaches in a principled way. Current resources could be improved in two ways. First, most approaches use variations of the original problem description and dataset developed by Ramírez and Geffner [2009], all of them derived from problems in the International Planning Competition (IPC). While IPC domains constitute challenging settings for planning implementations, these domains are not necessarily realistic or representative of goal recognition problems. Second, and unlike planning algorithms, many of the approaches surveyed lack an openly accessible reference implementation amenable to objective comparison. Thus, developing a unified benchmark in a common format using representative goal recognition problems, and making all efforts openly available constitute key targets for future efforts.

Acknowledgements
Felipe acknowledges support from CNPq with projects 407058/2018-4 (Universal) and 302773/2019-3 (PQ). Ramon acknowledges support from ERC Advanced Grant WhiteMech (No. 834228) and EU ICT-48 2020 TAILOR (No. 952215).

References


